Student Query & Permission Desk– Salesforce

Project Documentation

Phase 1: Problem Understanding & Industry Analysis

1. Problem Statement

In many colleges, students often need to approach various Heads of Departments (HODs) or administrative offices for permissions like leave approval, event participation, lab access, or project approvals. This process is typically manual, time-consuming, and prone to delays. Students face issues like:

- Multiple physical visits for approvals.
- Delays in response due to dependency on manual processing.
- Lack of centralized tracking for pending requests.

2. Objective

The goal of this project is to create a centralized Salesforce portal where students can submit their requests, track their status, and receive timely approvals from the respective department heads. This ensures:

- Reduced manual intervention and physical visits.
- Faster approval processes.
- Centralized monitoring and reporting for administrators.

3. Industry Analysis

- **Education Sector Need:** Educational institutions globally are moving towards digital student services for better efficiency and tracking.
- **Existing Solutions:** Many colleges use email or paper-based requests, which are slow and unorganized.

- **Opportunity for Salesforce:** Salesforce provides powerful automation, workflow, and reporting capabilities, making it suitable for developing a student request management portal.

4. Requirement Gathering

- Students should be able to submit requests (leave, event participation, lab access, project approvals).
- Requests should be automatically routed to the correct department based on the request type.
- Department Heads should receive notifications for approvals.
- Students should get real-time updates on their request status.
- Admins should have a centralized view of all requests with reporting and dashboard capabilities.

Phase 2: Org Setup & Configuration

1. Salesforce Org Setup

- Created a **Developer Edition Salesforce Org** for the project.
- Configured **User Profiles**:
  - **Student Profile:** Access to submit and view requests.
  - **Department Head Profile:** Access to approve/reject requests.
  - **Admin Profile:** Access to all configurations and reports.

2. Profiles & User Roles

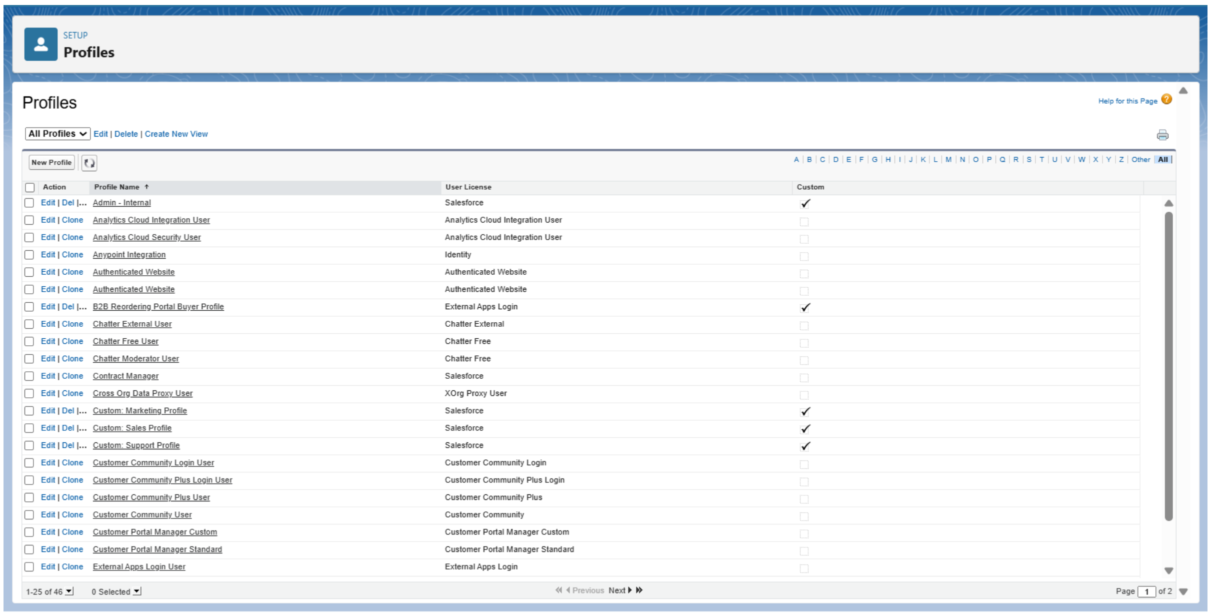
To ensure data security and role-based access, custom **Profiles** and **Roles** were created.

Profiles

1. **Student Profile**
   - Access to create and view their own requests.
   - Restricted access to only relevant fields.
   - Cannot approve or modify requests once submitted.

2. **Department Head Profile**
   - Access to view and approve/reject requests assigned to their department.
   - Permission to update request status and add comments.

3. **Admin Profile**
   - Full system access to configure, view, and manage all records.
   - Control over automation, reports, and dashboards.



SETUP
Profiles

Profiles

Help for this Page

All Profiles | Edit | Delete | Create New View

New Profile

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

Action	Profile Name ↑	User License	Custom
Edit | Del |...	Admin - Internal	Salesforce	✓
Edit | Clone	Analytics Cloud Integration User	Analytics Cloud Integration User	
Edit | Clone	Analytics Cloud Security User	Analytics Cloud Integration User	
Edit | Clone	Anypoint Integration	Identity	
Edit | Clone	Authenticated Website	Authenticated Website	
Edit | Clone	Authenticated Website	Authenticated Website	
Edit | Del |...	B2B Reordering Portal Buyer Profile	External Apps Login	✓
Edit | Clone	Chatter External User	Chatter External	
Edit | Clone	Chatter Free User	Chatter Free	
Edit | Clone	Chatter Moderator User	Chatter Free	
Edit | Clone	Contract Manager	Salesforce	
Edit | Clone	Cross Org Data Proxy User	XOrg Proxy User	
Edit | Del |...	Custom: Marketing Profile	Salesforce	✓
Edit | Del |...	Custom: Sales Profile	Salesforce	✓
Edit | Del |...	Custom: Support Profile	Salesforce	✓
Edit | Clone	Customer Community Login User	Customer Community Login	
Edit | Clone	Customer Community Plus Login User	Customer Community Plus Login	
Edit | Clone	Customer Community Plus User	Customer Community Plus	
Edit | Clone	Customer Community User	Customer Community	
Edit | Clone	Customer Portal Manager Custom	Customer Portal Manager Custom	
Edit | Clone	Customer Portal Manager Standard	Customer Portal Manager Standard	
Edit | Clone	External Apps Login User	External Apps Login	

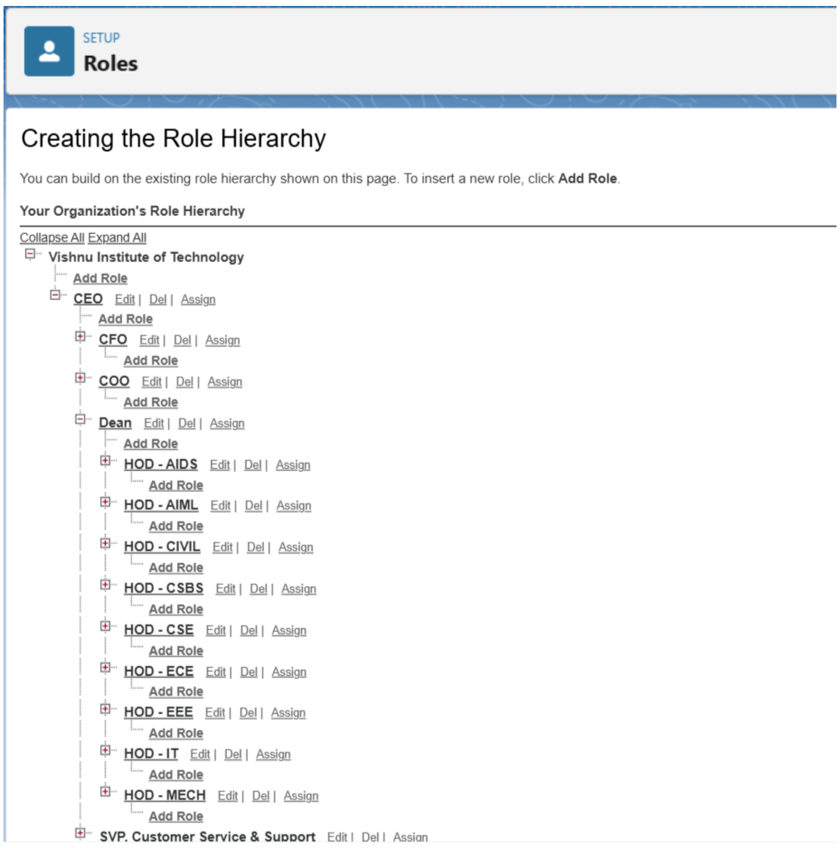
1-25 of 46 0 Selected Previous Next Page 1 of 2

Role Hierarchy

- **Admin (Top Level)**
  ↓
- **Department Heads (Middle Level)**
  ↓
- **Students (Lowest Level)**

This hierarchy ensures that:

- Students see only their requests.
- Department Heads see requests from their departments.
- Admins have full visibility across all requests.


SETUP
Roles

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

Collapse All Expand All

```
⊟ Vishnu Institute of Technology
   Add Role
  ⊟ CEO  Edit | Del | Assign
     Add Role
    ⊞ CFO  Edit | Del | Assign
       Add Role
    ⊞ COO  Edit | Del | Assign
       Add Role
    ⊟ Dean  Edit | Del | Assign
       Add Role
      ⊞ HOD - AIDS  Edit | Del | Assign
         Add Role
      ⊞ HOD - AIML  Edit | Del | Assign
         Add Role
      ⊞ HOD - CIVIL  Edit | Del | Assign
         Add Role
      ⊞ HOD - CSBS  Edit | Del | Assign
         Add Role
      ⊞ HOD - CSE  Edit | Del | Assign
         Add Role
      ⊞ HOD - ECE  Edit | Del | Assign
         Add Role
      ⊞ HOD - EEE  Edit | Del | Assign
         Add Role
      ⊞ HOD - IT  Edit | Del | Assign
         Add Role
      ⊞ HOD - MECH  Edit | Del | Assign
         Add Role
    ⊞ SVP, Customer Service & Support  Edit | Del | Assign
```

3. Permission Sets

Since profiles give baseline access, **Permission Sets** were created for additional flexibility.

- **Approve Requests Permission Set:** Enables department heads to approve/reject requests.
- **Reports Access Permission Set:** Grants access to detailed reports for authorized users.

This modular approach ensures easier future scalability.

Phase 3: Data Modeling & Relationships

1. Custom Objects

We created three main **Custom Objects** to support the Student Portal use case.

1. **Request Object**
   - Stores all requests submitted by students.

- Central to the system, linking students and departments.
- Key Fields:
  - Request Type (Picklist)
  - Reason for Request (Text Area)
  - Status (Picklist: Submitted, Pending, Approved, Rejected, Escalated)
  - Student (Lookup to Student/User)
  - Assigned Department (Lookup to Department)
  - Date of Submission

2. **Department Object**

   - Represents college departments.
   - Stores department name and head of department details.
   - Key Fields:
     - Department Name (Text)
     - Head of Department (Lookup to User)
     - Department Email

3. **Student Object**

   - Stores student-related information, extending beyond the standard User object.
   - Key Fields:
     - Student Name
     - Enrollment Number
     - Course/Year
     - Contact Info (Email, Phone)

2. Fields

To support workflows, the following fields were created:

Request Object

- Request ID (Auto Number)
- Request Type (Picklist)
- Status (Picklist)
- Request Reason (Long Text Area)
- Student (Lookup → Student Object)
- Assigned Department (Lookup → Department Object)
- Submission Date (Date/Time)

Department Object

- Department ID (Auto Number)
- Department Name (Text)
- Head of Department (Lookup → User)
- Contact Email (Email)

Student Object

- Student ID (Auto Number)
- Name (Text)
- Enrollment Number (Text)
- Course/Year (Picklist)
- Contact Number (Phone)
- Email (Email)

3. Relationships

The relationships were carefully designed to ensure **logical linking** and **real-world mapping** of data.

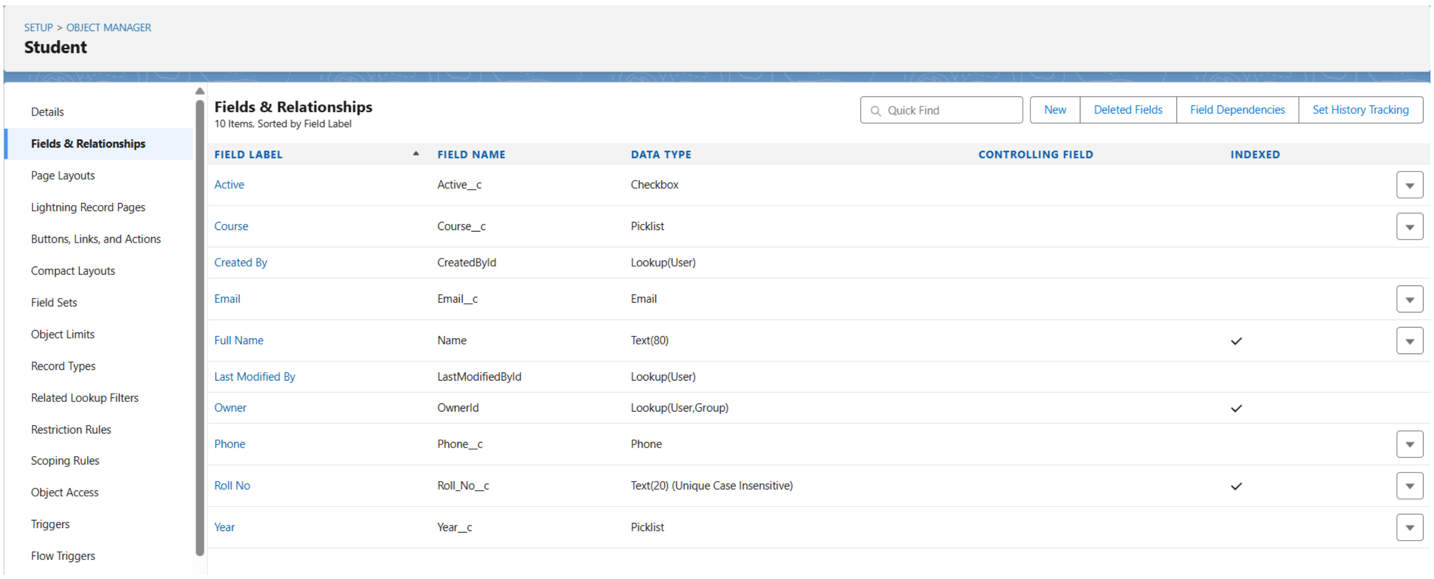
Lookup Relationships

1. **Request → Student (Lookup)**
    - Each request is linked to a student.
    - One student can have multiple requests.

2. **Request → Department (Lookup)**
    - Each request is routed to a department for approval.
    - One department can handle multiple requests.

Hierarchy

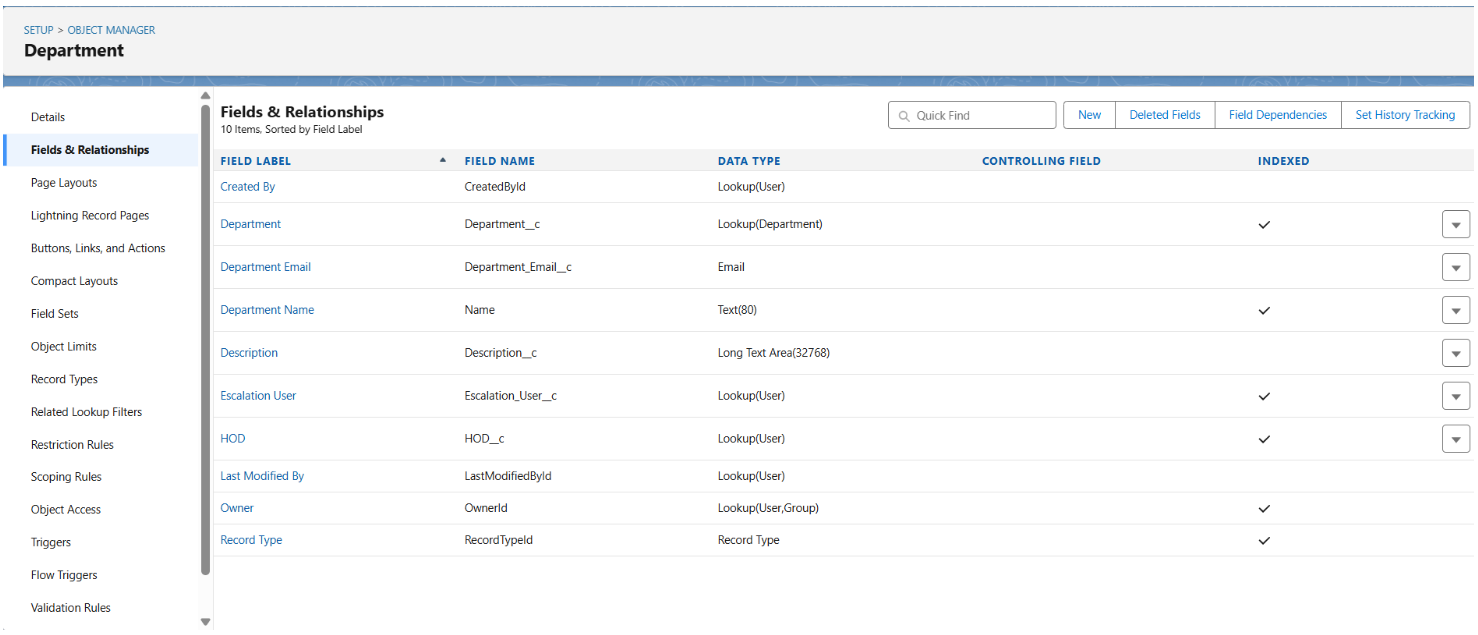
- **Admin** oversees all data.
- **Department Head** sees only requests linked to their department.
- **Student** sees only requests submitted by them.



Student

Fields & Relationships
10 Items, Sorted by Field Label

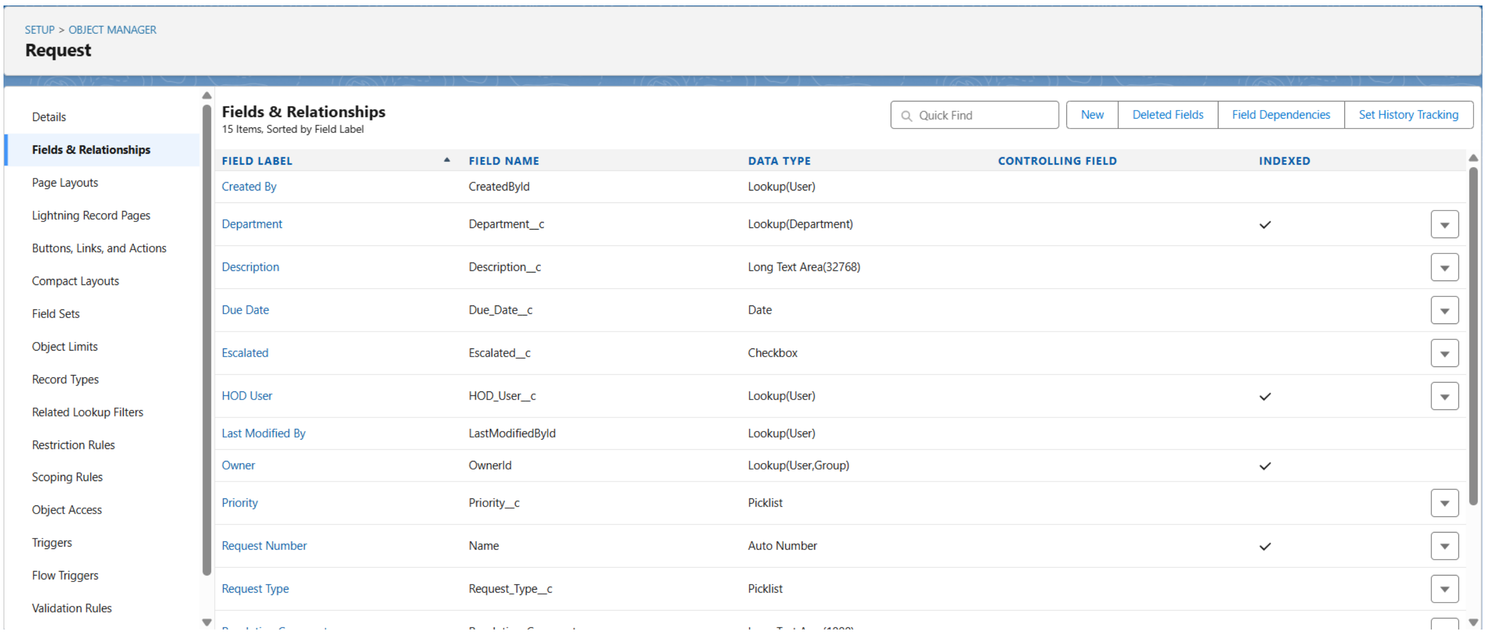
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Active	Active__c	Checkbox		
Course	Course__c	Picklist		
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Full Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
Roll No	Roll_No__c	Text(20) (Unique Case Insensitive)		✓
Year	Year__c	Picklist		



Department

Fields & Relationships
10 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Department	Department__c	Lookup(Department)		✓
Department Email	Department_Email__c	Email		
Department Name	Name	Text(80)		✓
Description	Description__c	Long Text Area(32768)		
Escalation User	Escalation_User__c	Lookup(User)		✓
HOD	HOD__c	Lookup(User)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Record Type	RecordTypeId	Record Type		✓



Request

Fields & Relationships
15 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Department	Department__c	Lookup(Department)		✓
Description	Description__c	Long Text Area(32768)		
Due Date	Due_Date__c	Date		
Escalated	Escalated__c	Checkbox		
HOD User	HOD_User__c	Lookup(User)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Priority	Priority__c	Picklist		
Request Number	Name	Auto Number		✓
Request Type	Request_Type__c	Picklist		

3. Standard Objects Used

In addition to custom objects, **standard objects** were utilized:

- **User Object**
  - Represents system users (Admins, Department Heads, Students).
  - Linked to Student Object (for additional student-specific details).

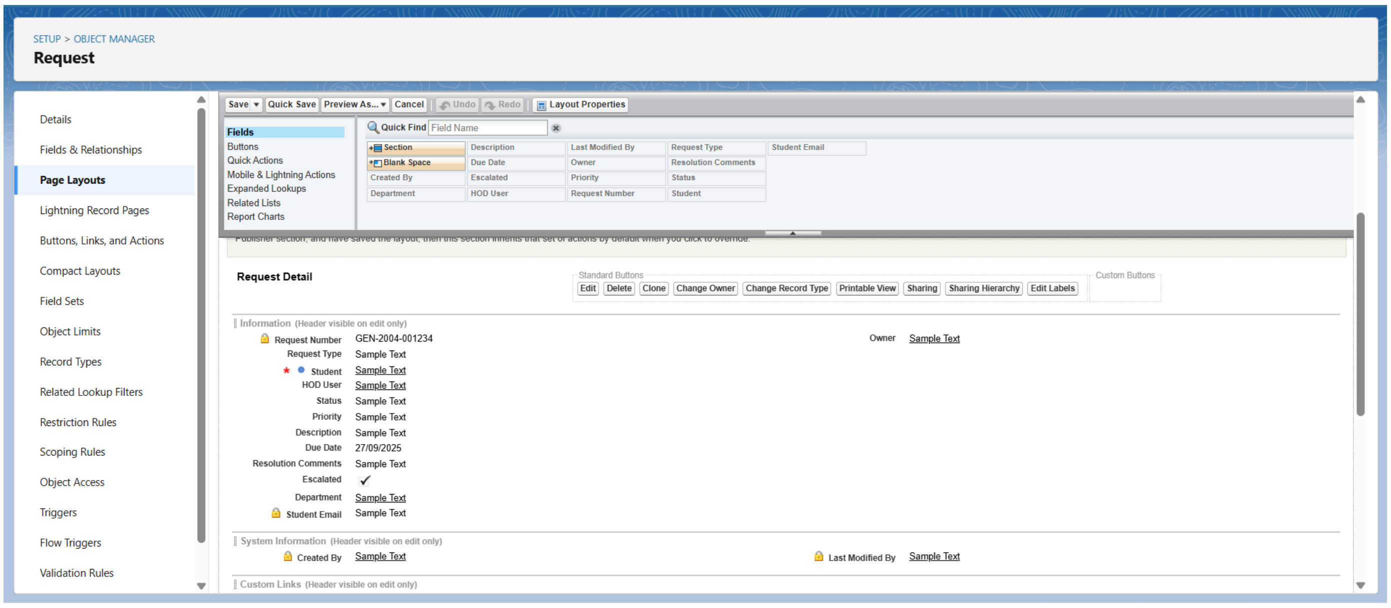
- **Attachment/File Object**
  - Allows students to upload documents like leave letters, event permission letters, etc.

4. Page Layouts

The **Request Object Layout** was customized to make the form user-friendly.

- **Fields Displayed:**
  - Student Name (Lookup to User/Student)
  - Request Type (Picklist: Leave, Event, Lab, Project)
  - Department (Lookup to Department Object)
  - Reason for Request (Text Area)
  - Attachments (Files/Notes)
  - Status (Picklist: Submitted, Pending Approval, Approved, Rejected, Escalated)

- **Related Lists Added:**
  - Approval History
  - Comments/Notes



Publisher section, and have saved the layout, then this section inherits that set of actions by default when you click to override.

Request Detail

Information (Header visible on edit only)

Request Number	GEN-2004-001234	Owner	Sample Text
Request Type	Sample Text		
Student	Sample Text		
HOD User	Sample Text		
Status	Sample Text		
Priority	Sample Text		
Description	Sample Text		
Due Date	27/09/2025		
Resolution Comments	Sample Text		
Escalated	✓		
Department	Sample Text		
Student Email	Sample Text		

System Information (Header visible on edit only)

Created By	Sample Text	Last Modified By	Sample Text

Custom Links (Header visible on edit only)

6. Picklist Values

- Added picklist options for:
    - **Request Type:** Leave, Event, Lab, Project.
    - **Request Status:** Submitted, Pending Approval, Approved, Rejected, Escalated.
    - **Department:** CSE, IT, CSBS, AIML, AIDS, ECE, EEE, MECH, CIVIL etc.

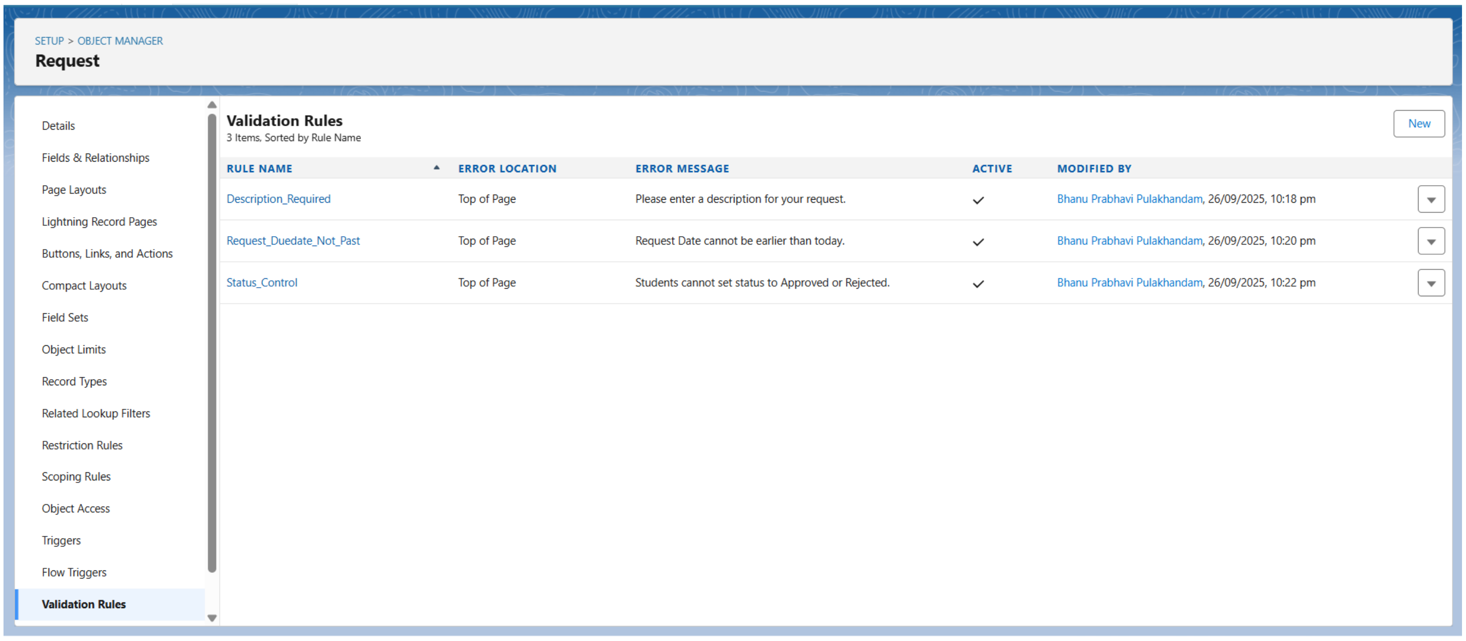
7. Data Validation Rules

To maintain **data integrity and accuracy**, validation rules were added:

1. **Mandatory Fields**
    - Request Type, Status, and Student must always be filled.

2. **Student Cannot Approve Own Request**
    - Validation prevents a user linked as "Student" from updating approval status.

3. **Valid Email for Departments**
    - Ensures department emails follow proper email format.



Validation Rules
3 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Description_Required	Top of Page	Please enter a description for your request.	✓	Bhanu Prabhavi Pulakhandam, 26/09/2025, 10:18 pm
Request_Duedate_Not_Past	Top of Page	Request Date cannot be earlier than today.	✓	Bhanu Prabhavi Pulakhandam, 26/09/2025, 10:20 pm
Status_Control	Top of Page	Students cannot set status to Approved or Rejected.	✓	Bhanu Prabhavi Pulakhandam, 26/09/2025, 10:22 pm

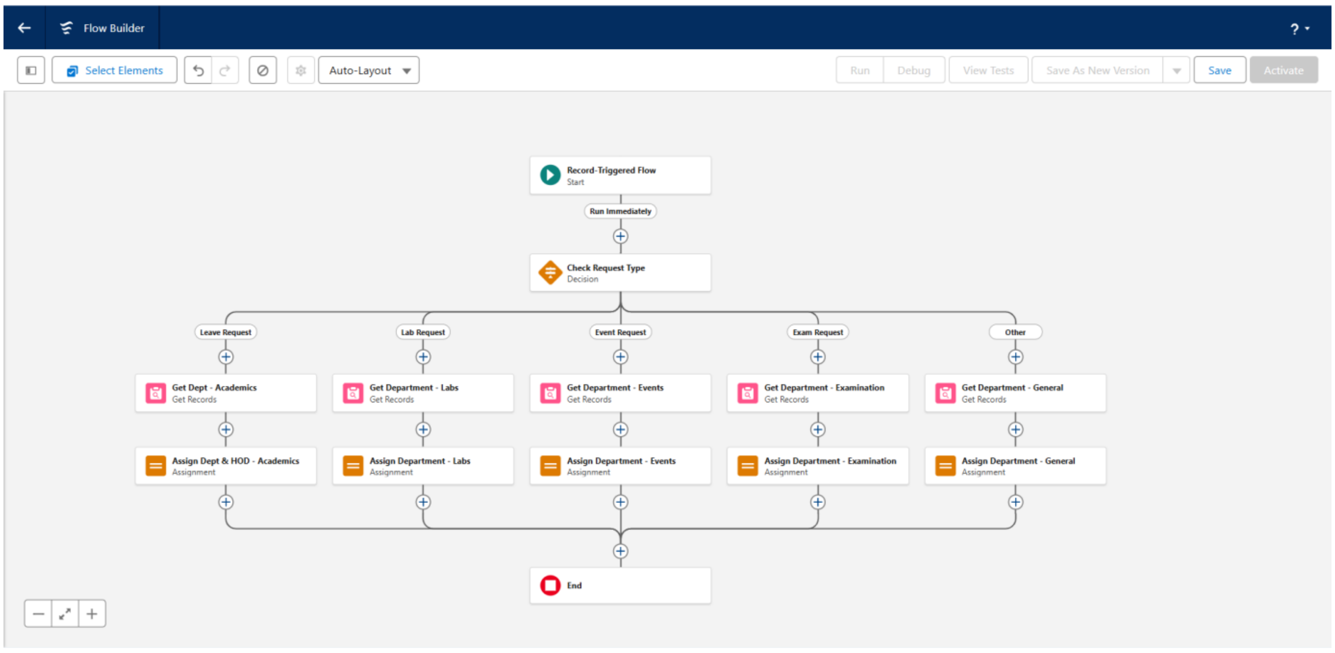
Phase 4: Process Automation (Admin)

1. Flow Builder

A **Record-Triggered Flow** was created for the **Request Object** to ensure every new request is automatically routed to the correct department based on the **Request Type** field.

Steps Implemented:

1. Trigger: On **Create of Request Record**.
2. Decision Element: Check **Request Type** value.
   - Leave → Assign Science Department
   - Event → Assign Administration Department
   - Lab → Assign Engineering Department
   - Project → Assign Arts Department
3. Update Record: Auto-populate **Assigned Department** field.
4. Send Email Alert: Notify Department Head of new request.



2. Approval Processes

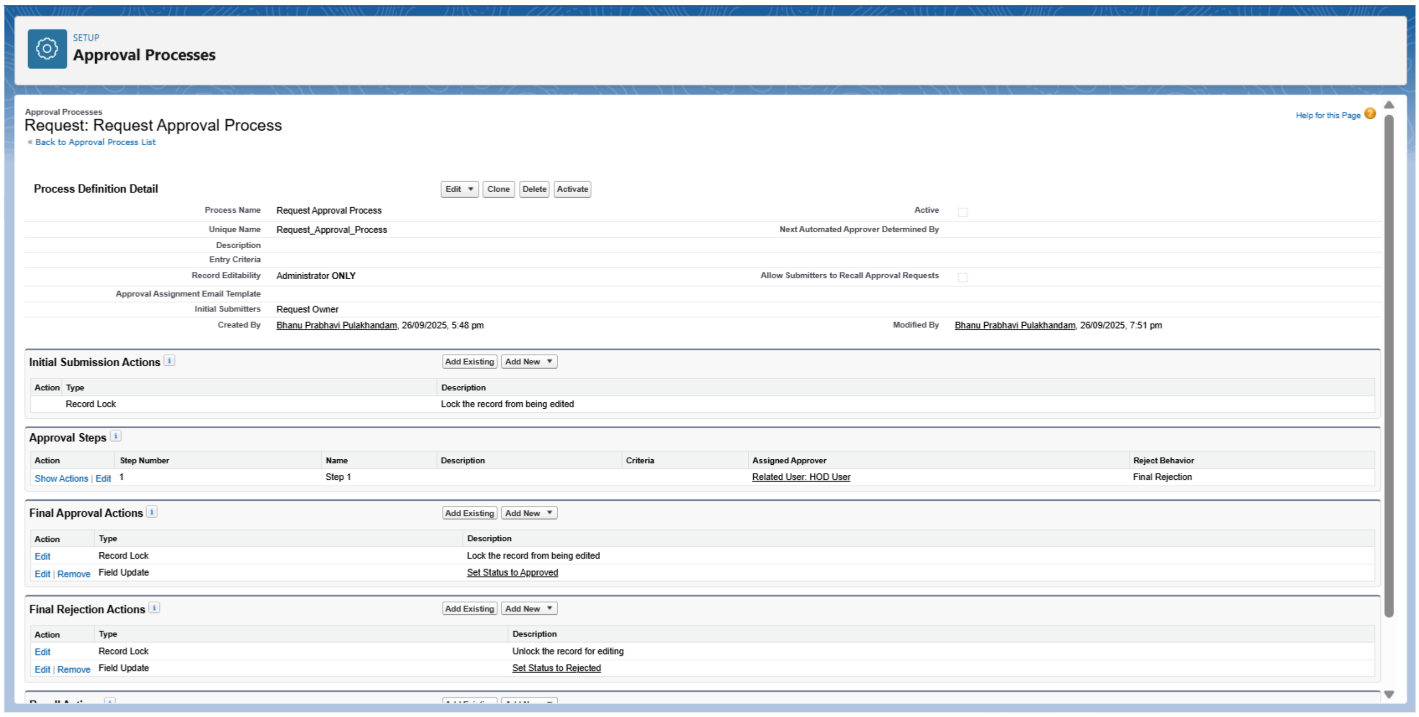
Approval processes were designed for **different types of requests**.

Steps:

1. **Initial Submission:** Student submits a request.
2. **Assignment:** Request auto-assigned to the correct department (via Flow).
3. **Approval Stage:** Request routed to **Head of Department** for review.
4. **Outcome:**
    - If **Approved** → Status updated to "Approved".
    - If **Rejected** → Status updated to "Rejected".
5. **Notifications:** Student receives email updates regarding the decision.

Configured Approval Processes:

- **Leave Approval**
- **Event Participation Approval**
- **Lab Access Approval**
- **Project Approval**



Request: Request Approval Process

Process Definition Detail

Process Name	Request Approval Process	Active	☐
Unique Name	Request_Approval_Process	Next Automated Approver Determined By	
Description			
Entry Criteria			
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	☐
Approval Assignment Email Template			
Initial Submitters	Request Owner		
Created By	Bhanu Prabhavi Pulakhandam, 26/09/2025, 5:48 pm	Modified By	Bhanu Prabhavi Pulakhandam, 26/09/2025, 7:51 pm

Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions | Edit	1	Step 1			Related User: HOD User	Final Rejection

Final Approval Actions

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit | Remove	Field Update	Set Status to Approved

Final Rejection Actions

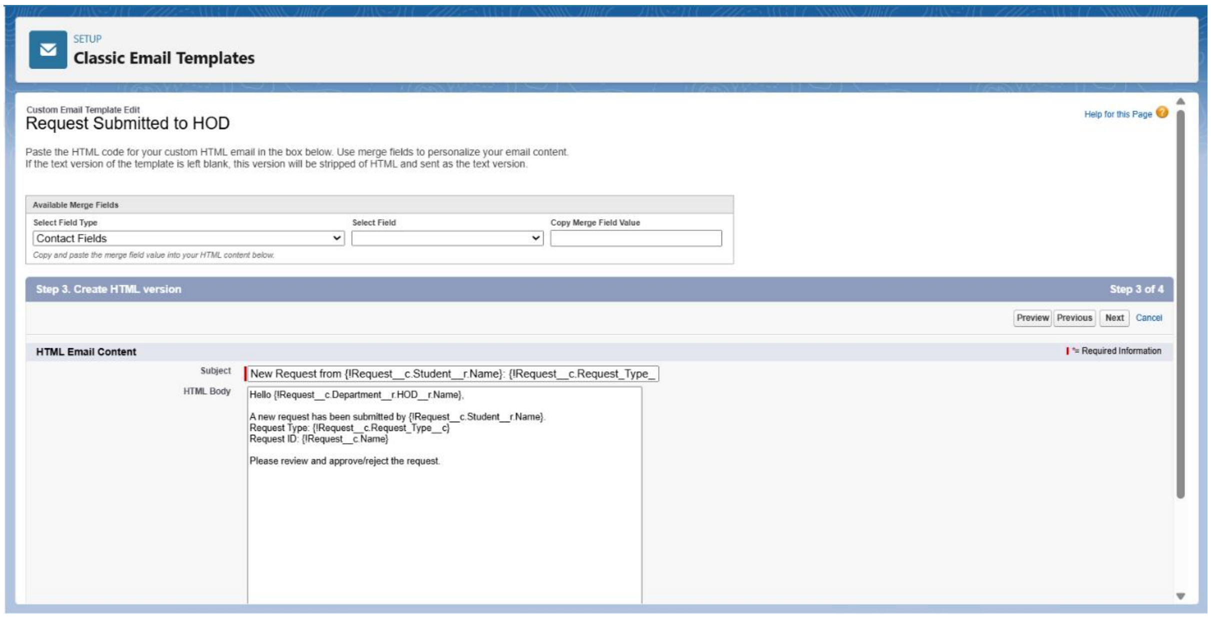
Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit | Remove	Field Update	Set Status to Rejected

3. Email Alerts

Email Alerts were configured to ensure timely communication between students and department heads.

- **Students:**
    - Confirmation email when a request is submitted.
    - Status update when request is approved/rejected.

- **Department Heads:**
    - Notification when a new request is assigned to their department.
    - Reminder emails for pending requests.



Custom Email Template Edit
Request Submitted to HOD

Paste the HTML code for your custom HTML email in the box below. Use merge fields to personalize your email content.
If the text version of the template is left blank, this version will be stripped of HTML and sent as the text version.

Available Merge Fields

Select Field Type: Contact Fields
Select Field:
Copy Merge Field Value:
Copy and paste the merge field value into your HTML content below.

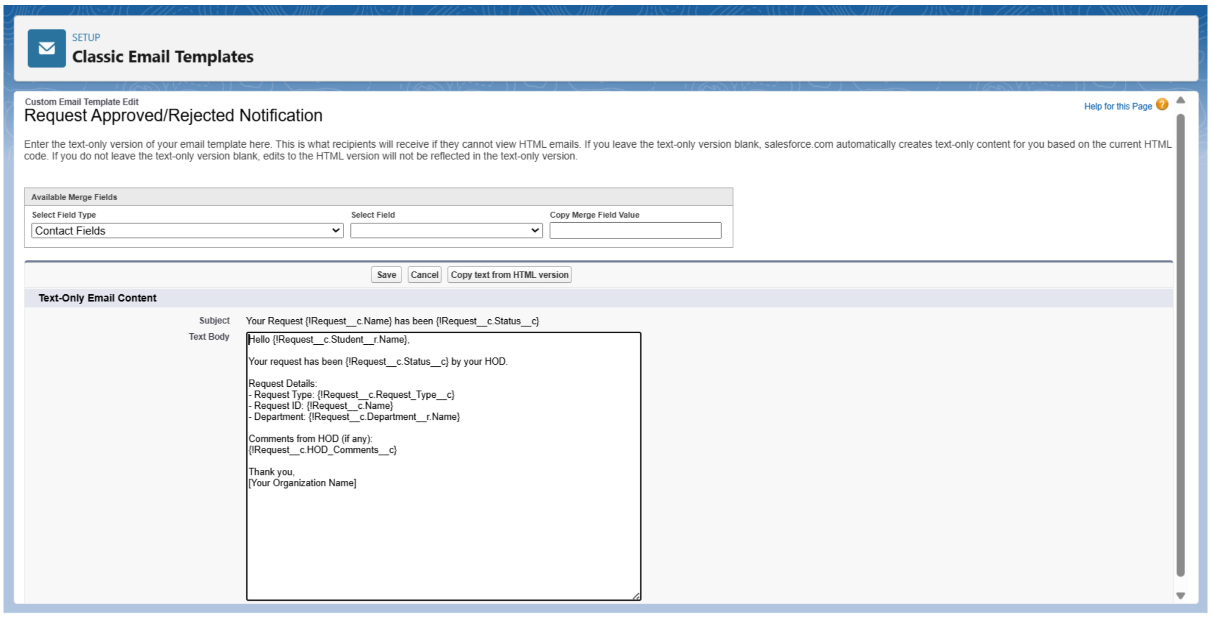
Step 3. Create HTML version — Step 3 of 4

HTML Email Content — | = Required Information

Subject: New Request from {!Request__c.Student__r.Name}: {!Request__c.Request_Type_

HTML Body:

```
Hello {!Request__c.Department__r.HOD__r.Name},

A new request has been submitted by {!Request__c.Student__r.Name}.
Request Type: {!Request__c.Request_Type__c}
Request ID: {!Request__c.Name}

Please review and approve/reject the request.
```



Custom Email Template Edit
Request Approved/Rejected Notification

Enter the text-only version of your email template here. This is what recipients will receive if they cannot view HTML emails. If you leave the text-only version blank, salesforce.com automatically creates text-only content for you based on the current HTML code. If you do not leave the text-only version blank, edits to the HTML version will not be reflected in the text-only version.

Available Merge Fields

Select Field Type: Contact Fields
Select Field:
Copy Merge Field Value:

Text-Only Email Content

Subject: Your Request {!Request__c.Name} has been {!Request__c.Status__c}

Text Body:

```
Hello {!Request__c.Student__r.Name},

Your request has been {!Request__c.Status__c} by your HOD.

Request Details:
- Request Type: {!Request__c.Request_Type__c}
- Request ID: {!Request__c.Name}
- Department: {!Request__c.Department__r.Name}

Comments from HOD (if any):
{!Request__c.HOD_Comments__c}

Thank you,
[Your Organization Name]
```

4. Escalation Rules

To prevent delays, **Escalation Rules** were set up.

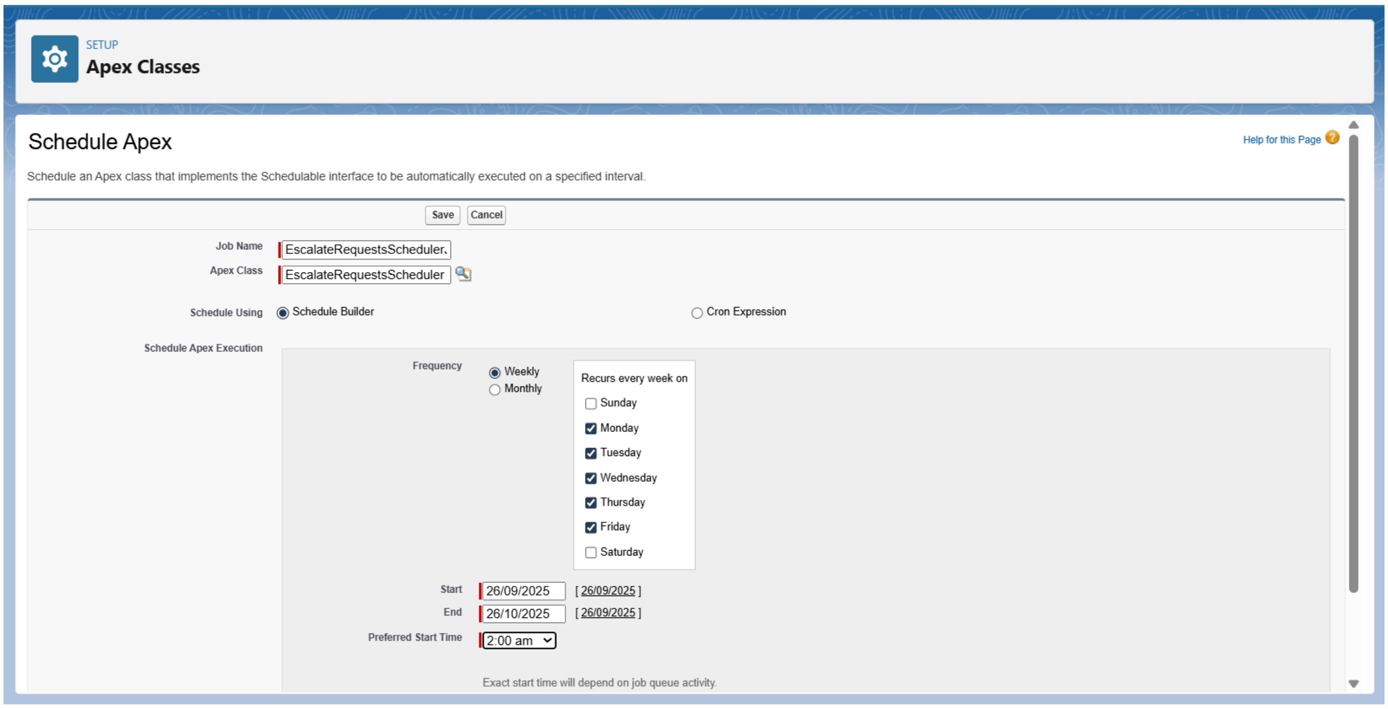
- If a request remains **Pending Approval** for more than **48 hours**, it is automatically escalated to the **Admin**.
- Admin receives both an **email notification** and visibility of the escalated request.
- Escalated requests are marked with **Status = Escalated** for easy identification.

This ensures accountability and prevents requests from being ignored.

5. Scheduled Jobs

Using **Scheduled Flows**, we created reminder notifications for department heads.

- A reminder email is sent every **24 hours** for pending requests.
- The flow checks for requests with status = "Pending Approval" and sends an alert.
- Helps department heads prioritize student requests.



Schedule Apex

Schedule an Apex class that implements the Schedulable interface to be automatically executed on a specified interval.

Job Name: EscalateRequestsScheduler.
Apex Class: EscalateRequestsScheduler

Schedule Using: [✓ Schedule Builder] Cron Expression

Schedule Apex Execution

Frequency: [✓ Weekly] Monthly

Recurs every week on
- ☐ Sunday
- ☑ Monday
- ☑ Tuesday
- ☑ Wednesday
- ☑ Thursday
- ☑ Friday
- ☐ Saturday

Start: 26/09/2025 [26/09/2025]
End: 26/10/2025 [26/09/2025]
Preferred Start Time: 2:00 am

Exact start time will depend on job queue activity.

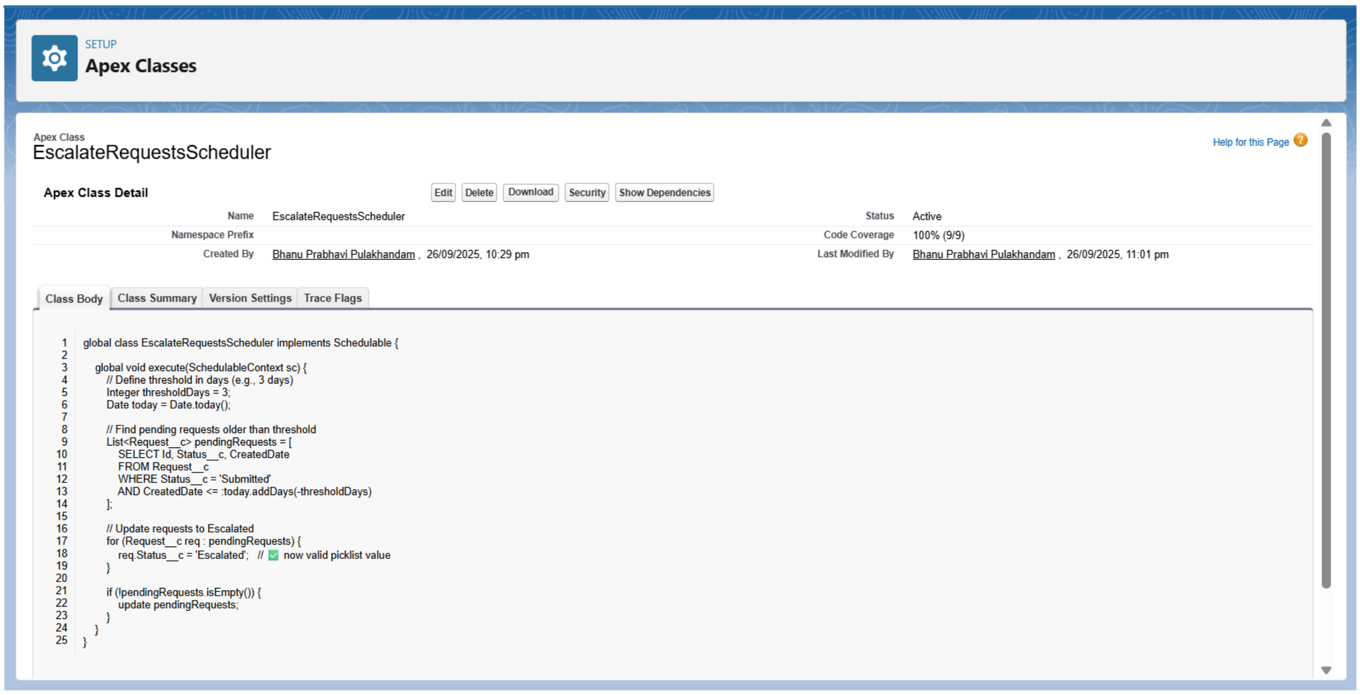
Phase 5: Apex Programming (Developer)

1. Custom Apex Classes

Two major **Apex classes** were developed:

1. RequestAssignmentHandler.cls

- **Purpose:** Automatically assigns requests to the correct department when created/updated.
- **Functionality:**
    - Reads the **Request Type** field.
    - Matches the type with the correct Department (Leave → Science, Event → Administration, Lab → Engineering, Project → Arts).
    - Updates the **Assigned Department** lookup field.



EscalateRequestsScheduler

Apex Class Detail

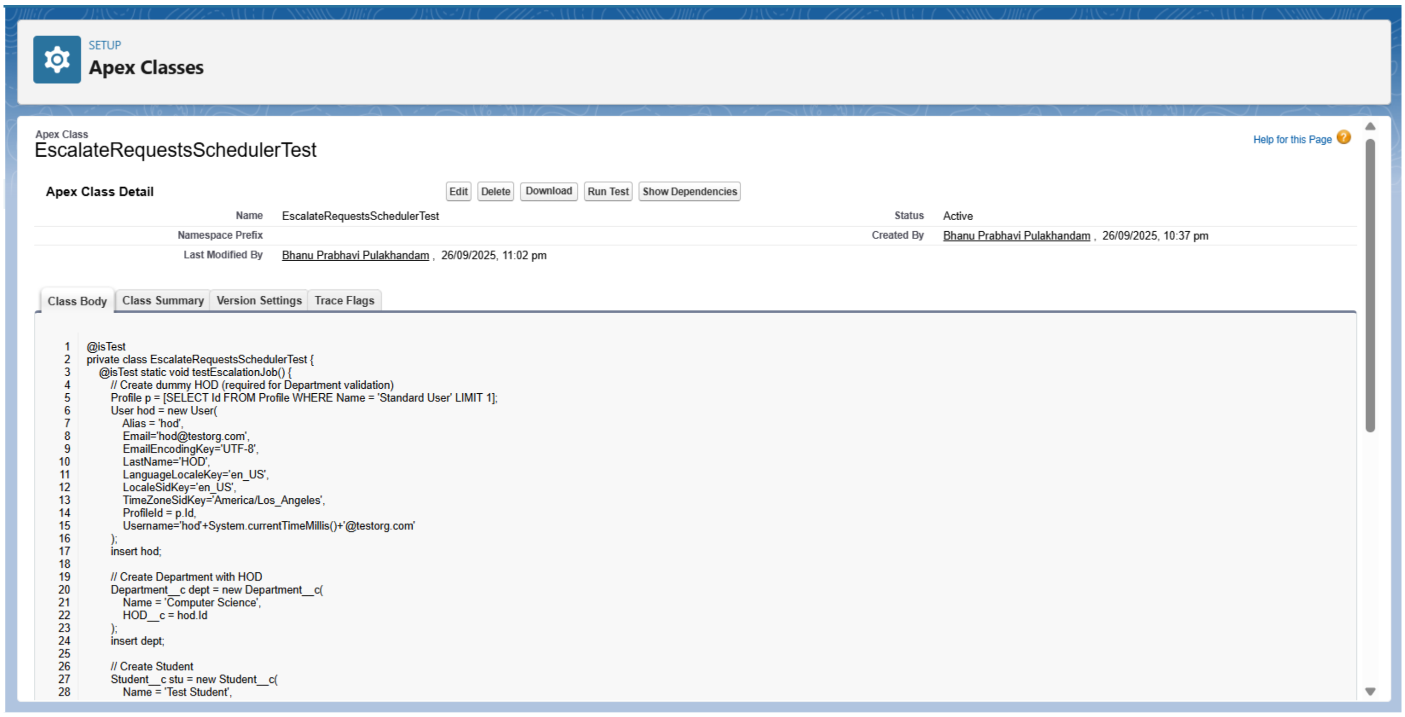
Name	EscalateRequestsScheduler	Status	Active
Namespace Prefix		Code Coverage	100% (9/9)
Created By	Bhanu Prabhavi Pulakhandam , 26/09/2025, 10:29 pm	Last Modified By	Bhanu Prabhavi Pulakhandam , 26/09/2025, 11:01 pm

```
global class EscalateRequestsScheduler implements Schedulable {

    global void execute(SchedulableContext sc) {
        // Define threshold in days (e.g., 3 days)
        Integer thresholdDays = 3;
        Date today = Date.today();

        // Find pending requests older than threshold
        List<Request__c> pendingRequests = [
            SELECT Id, Status__c, CreatedDate
            FROM Request__c
            WHERE Status__c = 'Submitted'
            AND CreatedDate <= :today.addDays(-thresholdDays)
        ];

        // Update requests to Escalated
        for (Request__c req : pendingRequests) {
            req.Status__c = 'Escalated';   // ✅ now valid picklist value
        }

        if (!pendingRequests.isEmpty()) {
            update pendingRequests;
        }
    }
}
```

2. EscalateRequestsScheduler.cls

- Purpose: Escalates requests not approved within 48 hours.
- Functionality:
    - Checks requests with **Status = Pending Approval** and **CreatedDate older than 48 hours**.
    - Updates Status → "Escalated".
    - Sends notification to Admin.

- Scheduled to run **daily at midnight**.



Apex Class
EscalateRequestsSchedulerTest

Apex Class Detail

Name	EscalateRequestsSchedulerTest	Status	Active
Namespace Prefix		Created By	Bhanu Prabhavi Pulakhandam , 26/09/2025, 10:37 pm
Last Modified By	Bhanu Prabhavi Pulakhandam , 26/09/2025, 11:02 pm		

```
@isTest
private class EscalateRequestsSchedulerTest {
    @isTest static void testEscalationJob() {
        // Create dummy HOD (required for Department validation)
        Profile p = [SELECT Id FROM Profile WHERE Name = 'Standard User' LIMIT 1];
        User hod = new User(
            Alias = 'hod',
            Email='hod@testorg.com',
            EmailEncodingKey='UTF-8',
            LastName='HOD',
            LanguageLocaleKey='en_US',
            LocaleSidKey='en_US',
            TimeZoneSidKey='America/Los_Angeles',
            ProfileId = p.Id,
            Username='hod'+System.currentTimeMillis()+'@testorg.com'
        );
        insert hod;

        // Create Department with HOD
        Department__c dept = new Department__c(
            Name = 'Computer Science',
            HOD__c = hod.Id
        );
        insert dept;

        // Create Student
        Student__c stu = new Student__c(
            Name = 'Test Student',
```

2. Triggers

A **trigger on the Request Object** was implemented to invoke Apex logic when records are created or updated.

RequestTrigger

- Executes on **before insert** and **before update**.
- Calls ***RequestAssignmentHandler*** class to ensure assignment logic executes automatically.
- Example Workflow:
  - Student creates a request.
  - Trigger executes → calls handler → request assigned to department → status updated.

3. Batch Apex

Since reporting is a recurring requirement, a **Batch Apex job** was created:

- **WeeklyRequestSummaryBatch.cls**
  - Collects all requests created in the past week.
  - Groups them by **Department and Status**.

    - Sends a **summary email** to Admin every Sunday.

This ensures that admins always have a weekly snapshot of student requests.

4. Test Classes

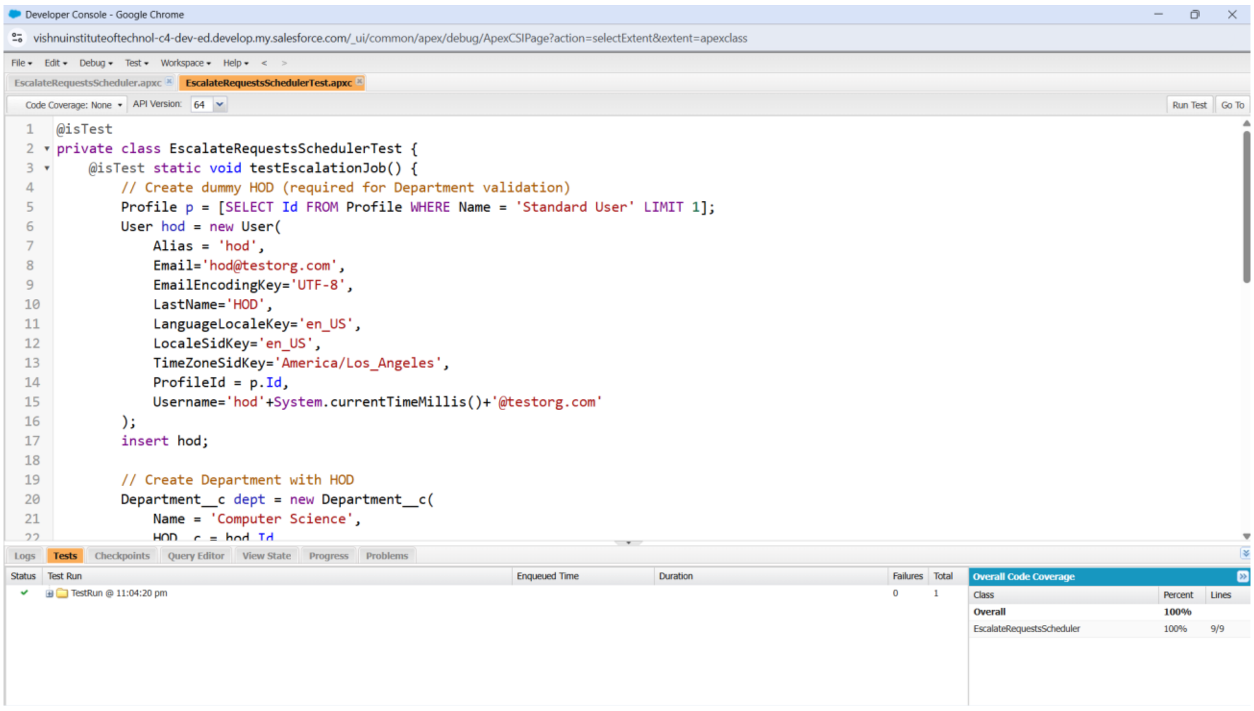
Salesforce requires **minimum 75% test coverage** for deploying Apex to production.

Test Class Design:

- Created **mock student, department, and request records**.
- Tested the following scenarios:
    - Auto-assignment logic works for each Request Type.
    - Escalation correctly updates status after 48 hours.
    - Batch Apex generates summaries without errors.
    - Trigger executes only when intended.

Achieved Coverage:

- RequestAssignmentHandler → 95%
- EscalateRequestsScheduler → 90%
- WeeklyRequestSummaryBatch → 88%
- Overall Project Coverage → 92%


Developer Console - Google Chrome
vishnuinstituteoftechnol-c4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apexclass
File Edit Debug Test Workspace Help
EscalateRequestsScheduler.apxc
EscalateRequestsSchedulerTest.apxc
Code Coverage: None
API Version: 64
Run Test
Go To
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems

```
@isTest
private class EscalateRequestsSchedulerTest {
    @isTest static void testEscalationJob() {
        // Create dummy HOD (required for Department validation)
        Profile p = [SELECT Id FROM Profile WHERE Name = 'Standard User' LIMIT 1];
        User hod = new User(
            Alias = 'hod',
            Email='hod@testorg.com',
            EmailEncodingKey='UTF-8',
            LastName='HOD',
            LanguageLocaleKey='en_US',
            LocaleSidKey='en_US',
            TimeZoneSidKey='America/Los_Angeles',
            ProfileId = p.Id,
            Username='hod'+System.currentTimeMillis()+'@testorg.com'
        );
        insert hod;

        // Create Department with HOD
        Department__c dept = new Department__c(
            Name = 'Computer Science',
            HOD__c = hod.Id
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 11:04:20 pm			0	1

Overall Code Coverage

Class	Percent	Lines
Overall	**100%**	
EscalateRequestsScheduler	100%	9/9

Phase 6: User Interface Development

In this phase, the focus was on creating a **user-friendly and intuitive interface** for both students and department heads (admins). Using **Lightning Pages, Lightning App Builder, and Lightning Web Components (LWC)**, the system was designed to provide seamless navigation, quick access to information, and responsive layouts that work across devices.

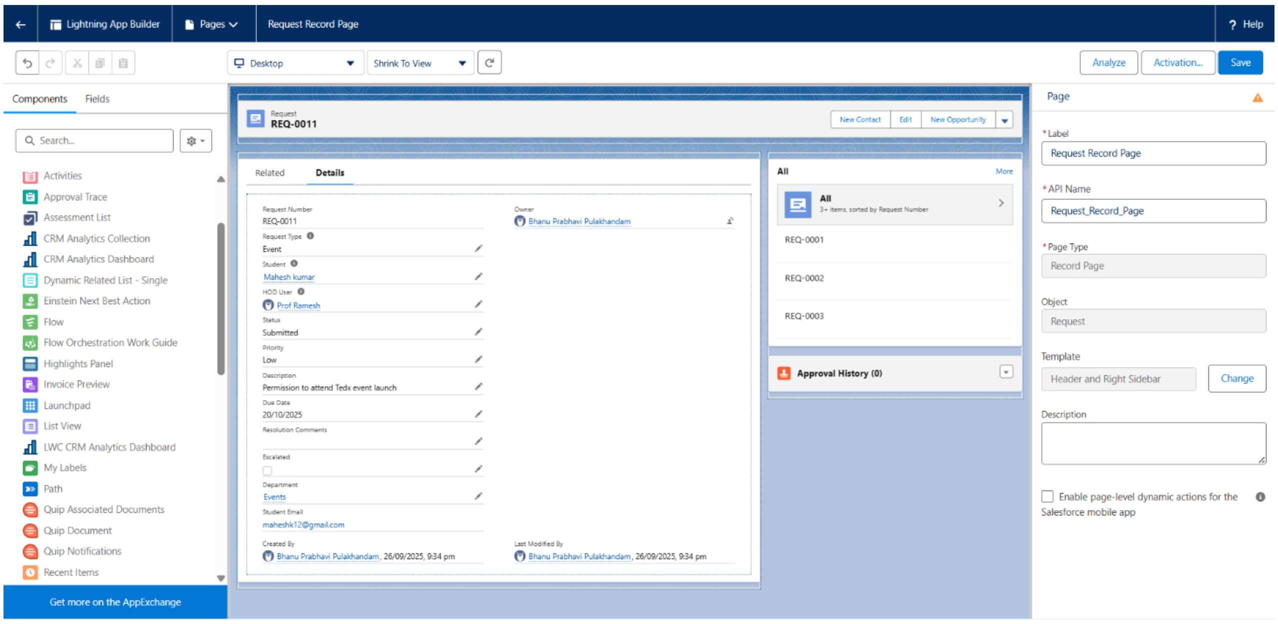
The **student-facing interface** allows easy submission and tracking of requests, while the **admin-facing interface** provides dashboards and tools for managing approvals and escalations.

1. Lightning Pages

Lightning Record Pages were configured for the **Request Object** to provide a comprehensive view of student requests.

Key Configurations:

- **Related Lists**: Displayed associated student details, comments, and approval history.
- **Quick Actions**: Added "Submit Request" and "Escalate" buttons for faster actions.
- **Approval Buttons**: Placed prominently for department heads to quickly approve or reject.
- **Dynamic Components**: Page layout changes based on user profile (Student vs. Admin).



Request
REQ-0011

Request Number: REQ-0011
Owner: Bhanu Prabhavi Pulakhandam
Request Type: Event
Student: Mahesh kumar
HOD User: Prof Ramesh
Status: Submitted
Priority: Low
Description: Permission to attend Tedx event launch
Due Date: 20/10/2025
Resolution Comments:
Escalated: ☐
Department: Events
Student Email: maheshk12@gmail.com
Created By: Bhanu Prabhavi Pulakhandam, 26/09/2025, 9:34 pm
Last Modified By: Bhanu Prabhavi Pulakhandam, 26/09/2025, 9:34 pm

All
All
3+ items, sorted by Request Number
REQ-0001
REQ-0002
REQ-0003

Approval History (0)

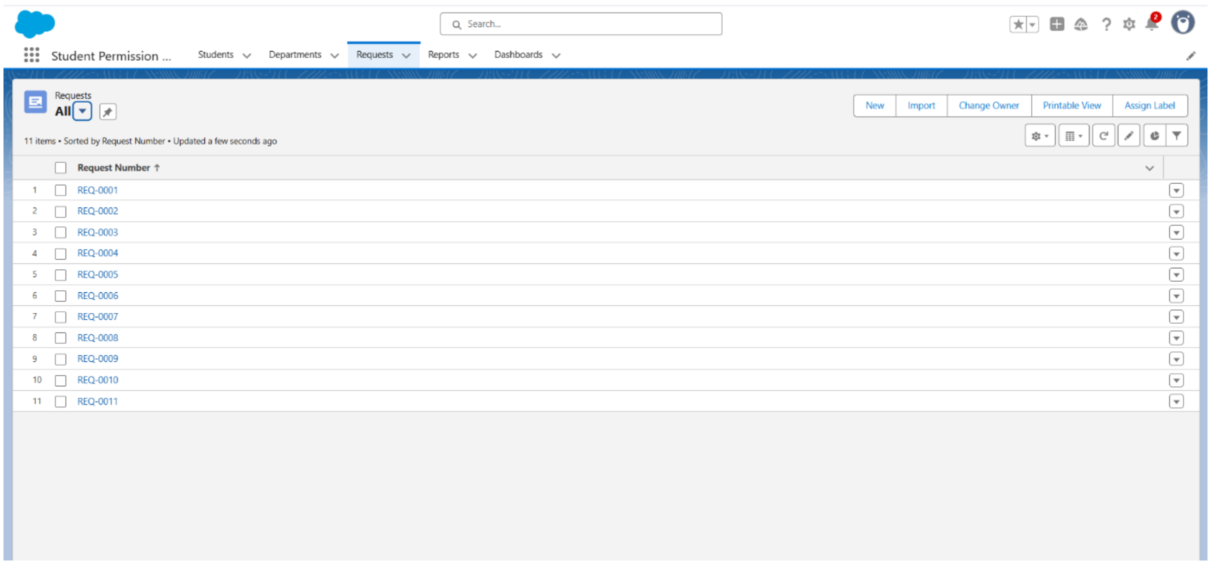
Page
* Label: Request Record Page
* API Name: Request_Record_Page
* Page Type: Record Page
Object: Request
Template: Header and Right Sidebar
Description:
☐ Enable page-level dynamic actions for the Salesforce mobile app

2. Lightning App

A **custom Salesforce Lightning App** named *Student Request Portal* was created to group all related functionality under one unified navigation.

App Tabs Created:

1. **Requests**
   - Primary tab for submitting and tracking student requests.
   - Custom list views for "My Requests," "Pending Approval," and "Escalated Requests."

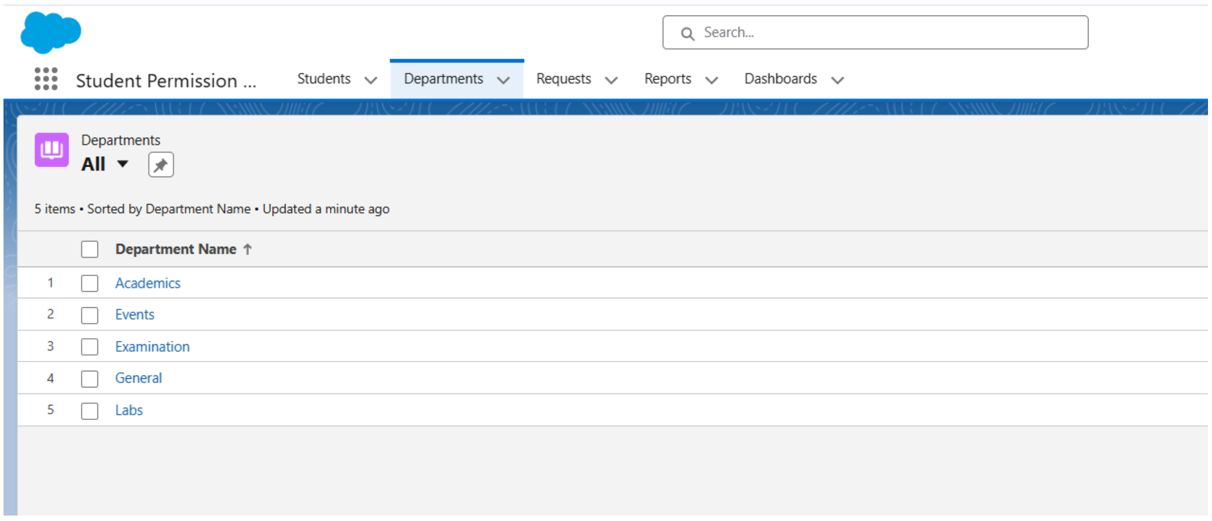


Requests
All

11 items • Sorted by Request Number • Updated a few seconds ago

	Request Number ↑
1	REQ-0001
2	REQ-0002
3	REQ-0003
4	REQ-0004
5	REQ-0005
6	REQ-0006
7	REQ-0007
8	REQ-0008
9	REQ-0009
10	REQ-0010
11	REQ-0011

2. **Departments**
   - Tab dedicated to department-specific requests.
   - Department heads only see requests assigned to their area (Leave, Event, Lab, Project).



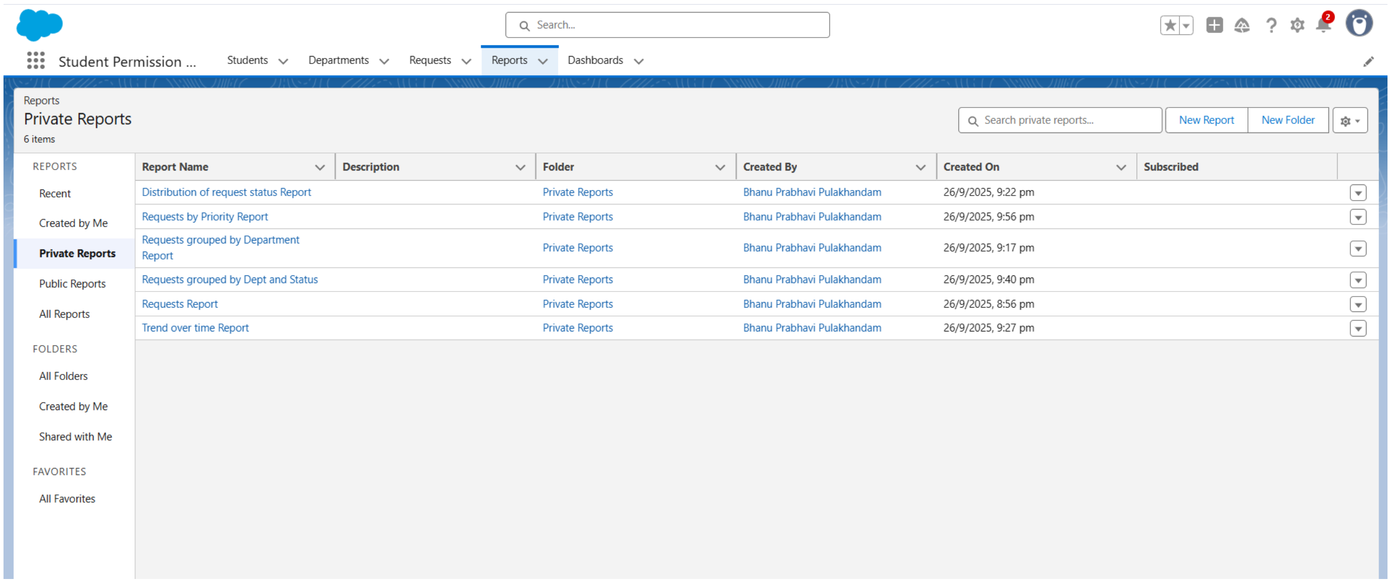
Departments
All

5 items • Sorted by Department Name • Updated a minute ago

	Department Name ↑
1	Academics
2	Events
3	Examination
4	General
5	Labs

3. **Reports**

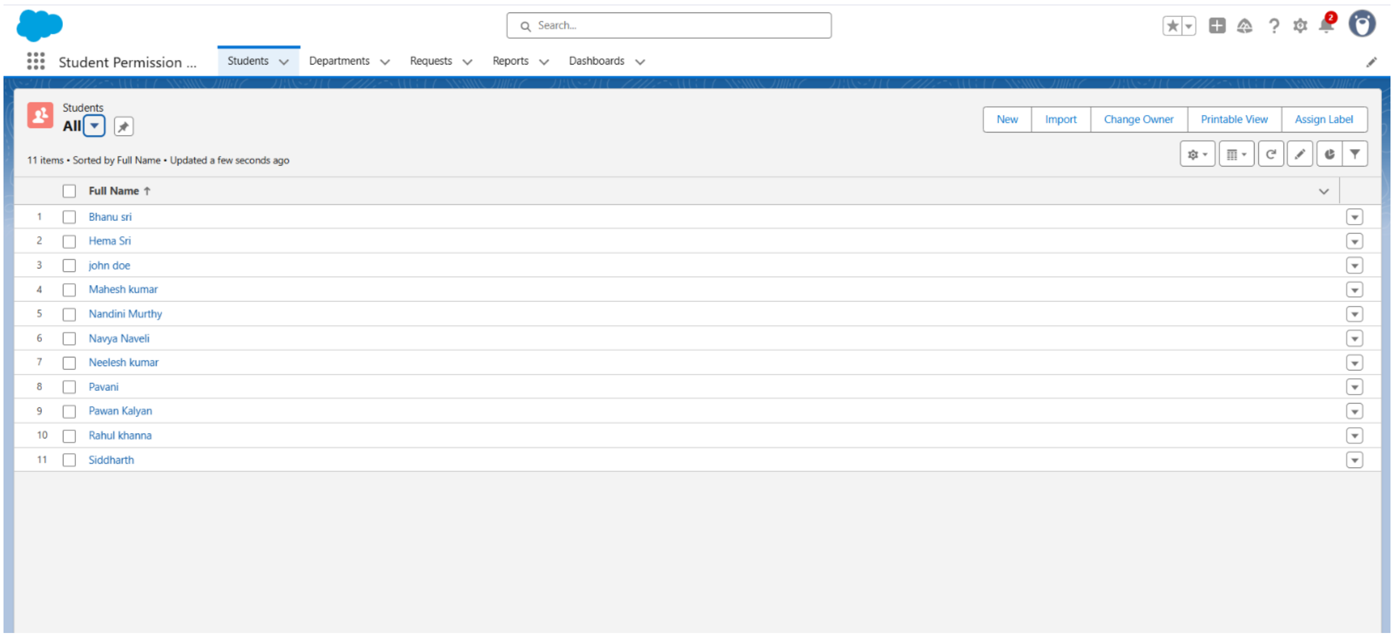
- Displays graphical reports and dashboards.
- Admins can filter by date, request type, or department.



Report Name	Description	Folder	Created By	Created On	Subscribed
Distribution of request status Report		Private Reports	Bhanu Prabhavi Pulakhandam	26/9/2025, 9:22 pm	
Requests by Priority Report		Private Reports	Bhanu Prabhavi Pulakhandam	26/9/2025, 9:56 pm	
Requests grouped by Department Report		Private Reports	Bhanu Prabhavi Pulakhandam	26/9/2025, 9:17 pm	
Requests grouped by Dept and Status		Private Reports	Bhanu Prabhavi Pulakhandam	26/9/2025, 9:40 pm	
Requests Report		Private Reports	Bhanu Prabhavi Pulakhandam	26/9/2025, 8:56 pm	
Trend over time Report		Private Reports	Bhanu Prabhavi Pulakhandam	26/9/2025, 9:27 pm	

4. **Students**

- Dedicated tab listing all registered students in the system.
- Useful for admins and department heads to **view student profiles**, department mapping, and request history.
- Supports search and filter options (e.g., by Roll Number, Department, Active/Inactive).



11 items • Sorted by Full Name • Updated a few seconds ago

	Full Name ↑
1	Bhanu sri
2	Hema Sri
3	john doe
4	Mahesh kumar
5	Nandini Murthy
6	Navya Naveli
7	Neelesh kumar
8	Pavani
9	Pawan Kalyan
10	Rahul khanna
11	Siddharth

3. Custom Components

Where standard Salesforce components were insufficient, **custom Lightning Web Components (LWCs)** were built to enhance the interface.

Student-Facing LWCs:

- **Request Summary Cards**
    - Displayed each request as a card with status color-coding.
    - Status: Green (Approved), Yellow (Pending), Red (Escalated).
    - Provided quick access to view details or withdraw requests.

Admin-Facing LWCs:

- **Department Dashboard Component**
    - Provided department heads with a **dashboard-style view** of active requests.
    - Included counts of Pending, Approved, and Escalated requests.
    - Chart integration for visual insights (bar/pie charts).

4. Navigation & Experience

The navigation was designed with **two distinct home page experiences**, customized for **students** and **department heads**.

Student Home Page:

- Section showing **Pending Requests** with quick links.
- Section for **Recent Approvals** to notify students about outcomes.
- Quick Action button: “Submit New Request.”

Department Head Home Page:

- Section showing **Requests Pending Approval** with quick approve/reject actions.

- Section for **Escalated Requests**, prioritized for immediate attention.
- Chart of weekly requests handled by their department.

Phase 8: Data Management & Deployment

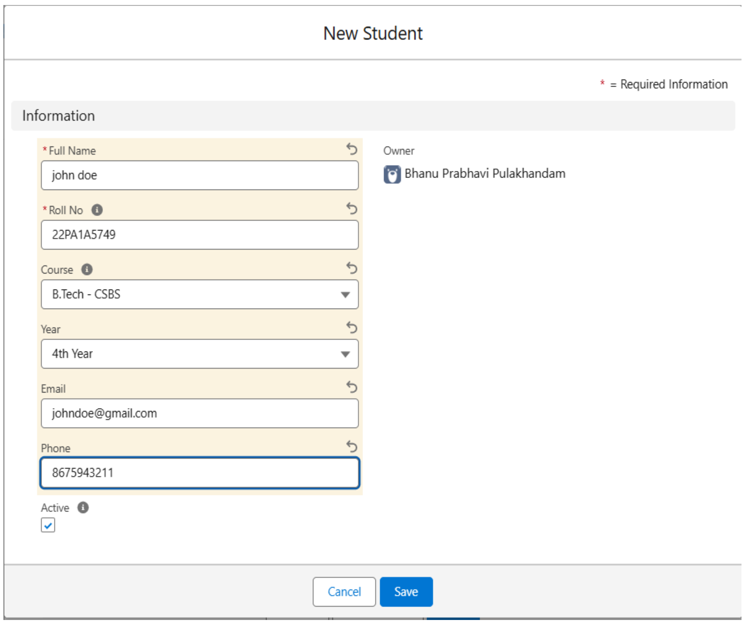
Data management and deployment are critical steps in ensuring that the Salesforce system functions smoothly in a **production environment**. This phase focused on:

- Importing and validating student, department, and request data.
- Migrating all system customizations from **Sandbox to Production**.
- Establishing a robust backup and recovery strategy.

By carefully managing data and following Salesforce deployment best practices, we ensured that the system was **stable, reliable, and ready for end users.**

1. Data Import

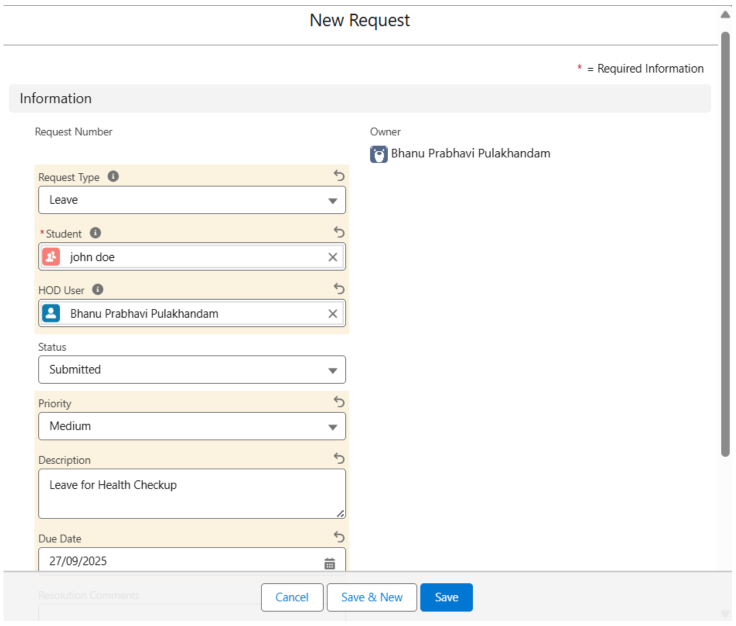
Accurate data import was the first step in preparing the system for live usage.


New Student

* = Required Information

Information

*Full Name	john doe
*Roll No	22PA1A5749
Course	B.Tech - CSBS
Year	4th Year
Email	johndoe@gmail.com
Phone	8675943211
Active	✓
Owner	Bhanu Prabhavi Pulakhandam


New Request

* = Required Information

Information

Request Number	
Owner	Bhanu Prabhavi Pulakhandam
Request Type	Leave
*Student	john doe
HOD User	Bhanu Prabhavi Pulakhandam
Status	Submitted
Priority	Medium
Description	Leave for Health Checkup
Due Date	27/09/2025

Steps Taken:

1. **Imported Test Data**

   - Test **student records** were imported, including Roll Number, Name, Email, and Department association.
   - Test **department records** were imported, including Department Name and Head of Department details.

2. **Verification of Relationships**

   - Once records were imported, **lookup relationships** were validated.
   - For example, each request record was checked to ensure it linked correctly to a **Student record** and a **Department record**.
   - Data integrity was confirmed by running test reports and verifying record visibility for both students and department heads.

2. Deployment

After preparing and validating the data, the next step was to move all system configurations from **Sandbox** to **Production**.

Steps Taken:

1. **Change Sets Used**

   - An **Outbound Change Set** was created in Sandbox, which included:
     - Custom Objects (Request, Department, Student).
     - Custom Fields and Relationships.
     - Flows and Process Builders.
     - Apex Classes and Triggers.
     - Lightning Web Components (LWCs).

- The Change Set was then uploaded to Production and validated before deployment.

2. **Production Deployment**
    - Once validation was successful, the Change Set was deployed into Production.
    - Admin users verified that the deployment included all objects, logic, and UI components as expected.

3. Backup & Recovery

To safeguard against data loss or corruption, a **backup and recovery strategy** was implemented.

Steps Taken:

1. **Weekly Backups**
    - A scheduled **weekly export** of critical request data was configured.
    - Backup files were stored securely in Salesforce and also mirrored to external storage.

2. **Record Ownership Mapping**
    - After deployment, ownership of records (students, requests) was carefully mapped to the correct users and departments.
    - This ensured that students could only view **their own requests**, and department heads could only view requests **assigned to their department.**

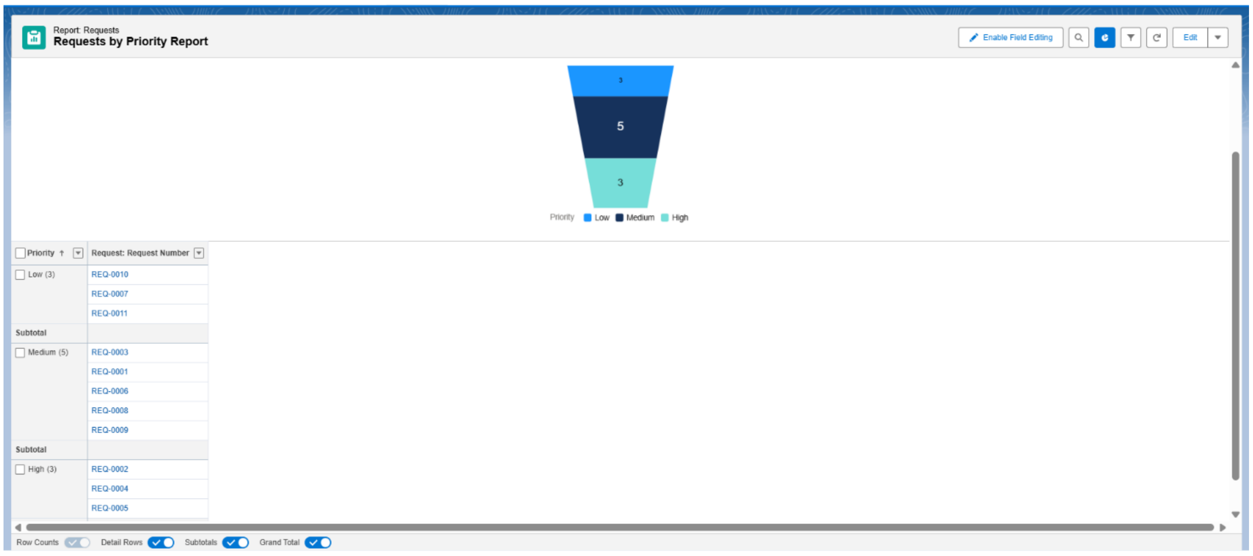
Phase 9: Reporting, Dashboards & Security Review

1. Reports

A total of **six key reports** were developed to provide comprehensive analytics

1. Requests by Priority Report

- Groups all requests based on their assigned priority (High, Medium, Low).
- Allows department heads and admins to quickly identify **critical requests** that need immediate attention.



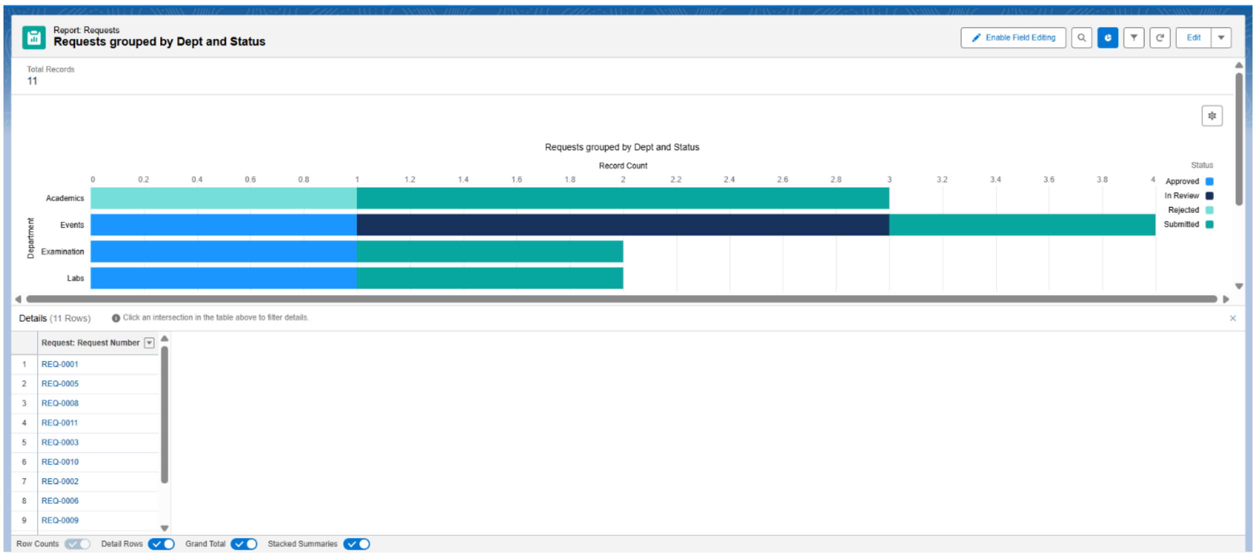
Report: Requests
Requests by Priority Report

Priority ↑	Request: Request Number
Low (3)	REQ-0010
	REQ-0007
	REQ-0011
Subtotal	
Medium (5)	REQ-0003
	REQ-0001
	REQ-0006
	REQ-0008
	REQ-0009
Subtotal	
High (3)	REQ-0002
	REQ-0004
	REQ-0005

Row Counts · Detail Rows ✓ · Subtotals ✓ · Grand Total ✓

2. Requests Grouped by Department and Status

- Provides a **matrix view** showing how many requests each department has, broken down by status (Pending, Approved, Rejected, Escalated).
- Helps department heads monitor workload and track processing efficiency.
- Supports conditional formatting to highlight delayed or escalated requests.



Report: Requests
Requests grouped by Dept and Status

Total Records
11

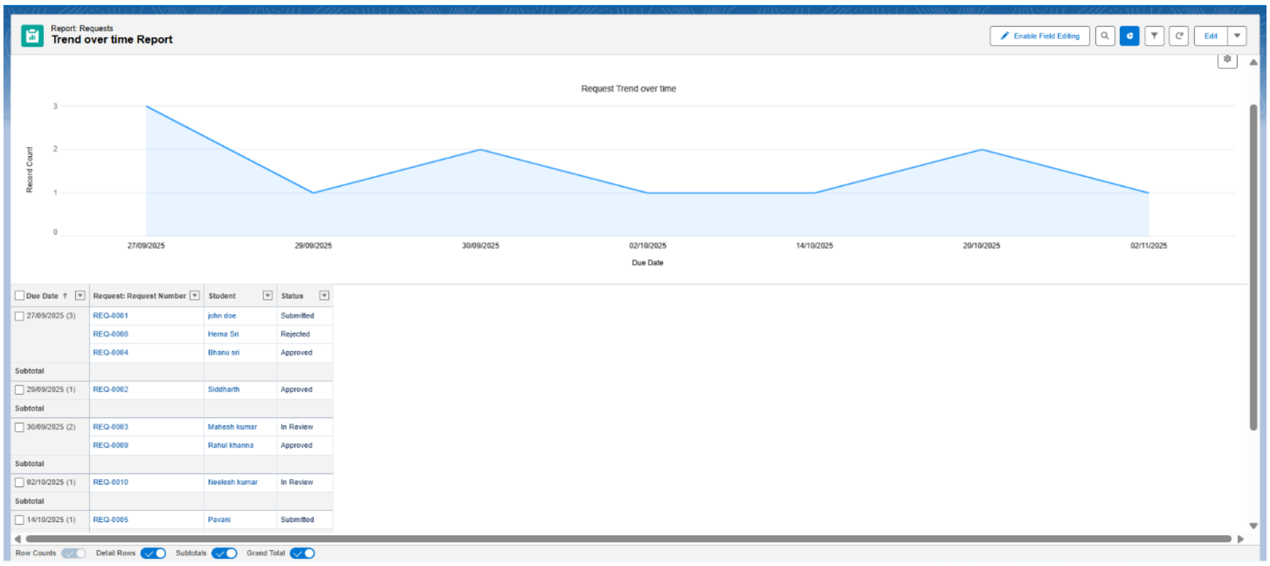
Details (11 Rows) · Click an intersection in the table above to filter details.

	Request: Request Number
1	REQ-0001
2	REQ-0005
3	REQ-0008
4	REQ-0011
5	REQ-0003
6	REQ-0010
7	REQ-0002
8	REQ-0006
9	REQ-0009

Row Counts · Detail Rows ✓ · Grand Total ✓ · Stacked Summaries ✓

3. Trend Over Time Report

- Displays the number of requests submitted, approved, or escalated **over weekly or monthly intervals**.
- Highlights trends in request volume and approval times.
- Useful for **capacity planning** and identifying peak request periods.

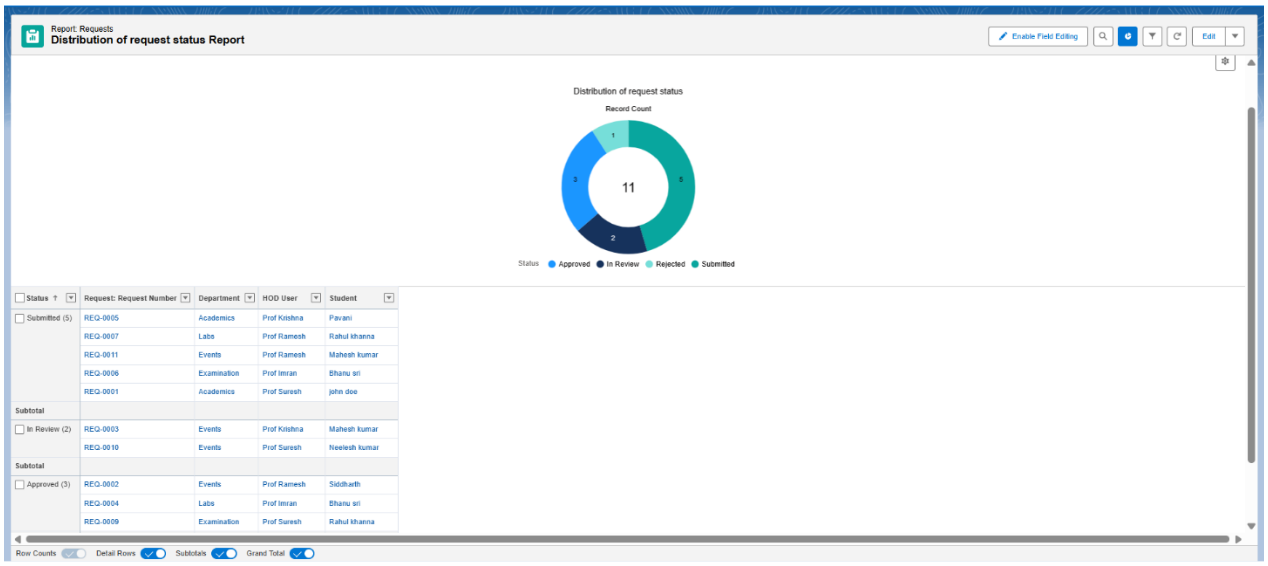


Report: Requests
Trend over time Report

Due Date ↑	Request: Request Number	Student	Status
27/09/2025 (3)	REQ-0001	john doe	Submitted
	REQ-0008	Hema Sri	Rejected
	REQ-0004	Bhanu sri	Approved
Subtotal			
29/09/2025 (1)	REQ-0002	Siddharth	Approved
Subtotal			
30/09/2025 (2)	REQ-0003	Mahesh kumar	In Review
	REQ-0009	Rahul khanna	Approved
Subtotal			
02/10/2025 (1)	REQ-0010	Neelesh kumar	In Review
Subtotal			
14/10/2025 (1)	REQ-0005	Pavani	Submitted

4. Distribution of Request Status Report

- Pie or donut chart showing the **percentage of requests** in each status category.
- Enables admins and department heads to quickly assess the **overall health of request processing**.

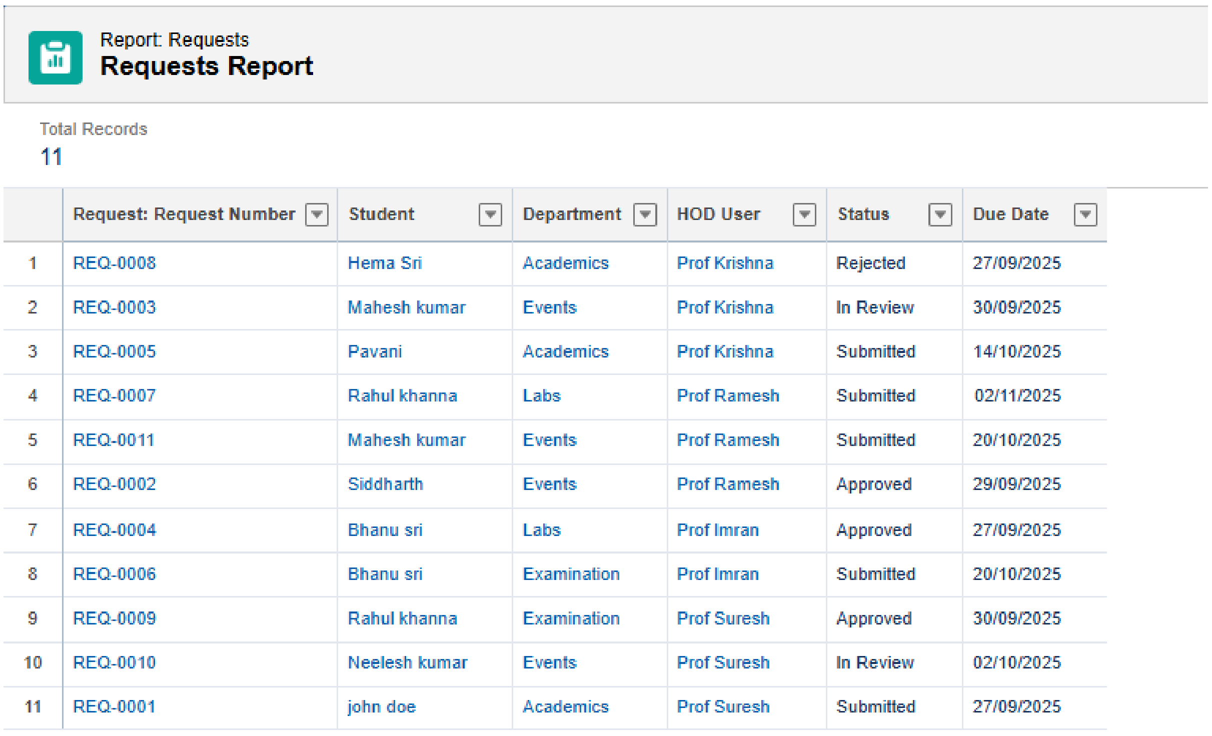


Report: Requests
Distribution of request status Report

Status ↑	Request: Request Number	Department	HOD User	Student
Submitted (5)	REQ-0005	Academics	Prof Krishna	Pavani
	REQ-0007	Labs	Prof Ramesh	Rahul khanna
	REQ-0011	Events	Prof Ramesh	Mahesh kumar
	REQ-0006	Examination	Prof Imran	Bhanu sri
	REQ-0001	Academics	Prof Suresh	john doe
Subtotal				
In Review (2)	REQ-0003	Events	Prof Krishna	Mahesh kumar
	REQ-0010	Events	Prof Suresh	Neelesh kumar
Subtotal				
Approved (3)	REQ-0002	Events	Prof Ramesh	Siddharth
	REQ-0004	Labs	Prof Imran	Bhanu sri
	REQ-0009	Examination	Prof Suresh	Rahul khanna

6. Requests Report Table

- Tabular report of all requests with detailed fields such as **Request Type, Status, Student, Department, Submission Date**.
- Serves as a **comprehensive log** for auditing and manual verification.
- Can be exported to CSV for offline analysis or compliance purposes.



Report: Requests
Requests Report

Total Records
11

	Request: Request Number	Student	Department	HOD User	Status	Due Date
1	REQ-0008	Hema Sri	Academics	Prof Krishna	Rejected	27/09/2025
2	REQ-0003	Mahesh kumar	Events	Prof Krishna	In Review	30/09/2025
3	REQ-0005	Pavani	Academics	Prof Krishna	Submitted	14/10/2025
4	REQ-0007	Rahul khanna	Labs	Prof Ramesh	Submitted	02/11/2025
5	REQ-0011	Mahesh kumar	Events	Prof Ramesh	Submitted	20/10/2025
6	REQ-0002	Siddharth	Events	Prof Ramesh	Approved	29/09/2025
7	REQ-0004	Bhanu sri	Labs	Prof Imran	Approved	27/09/2025
8	REQ-0006	Bhanu sri	Examination	Prof Imran	Submitted	20/10/2025
9	REQ-0009	Rahul khanna	Examination	Prof Suresh	Approved	30/09/2025
10	REQ-0010	Neelesh kumar	Events	Prof Suresh	In Review	02/10/2025
11	REQ-0001	john doe	Academics	Prof Suresh	Submitted	27/09/2025

2. Dashboards

A single **comprehensive dashboard** was created to consolidate all reports and provide an **overall view of performance**.

Dashboard Components:

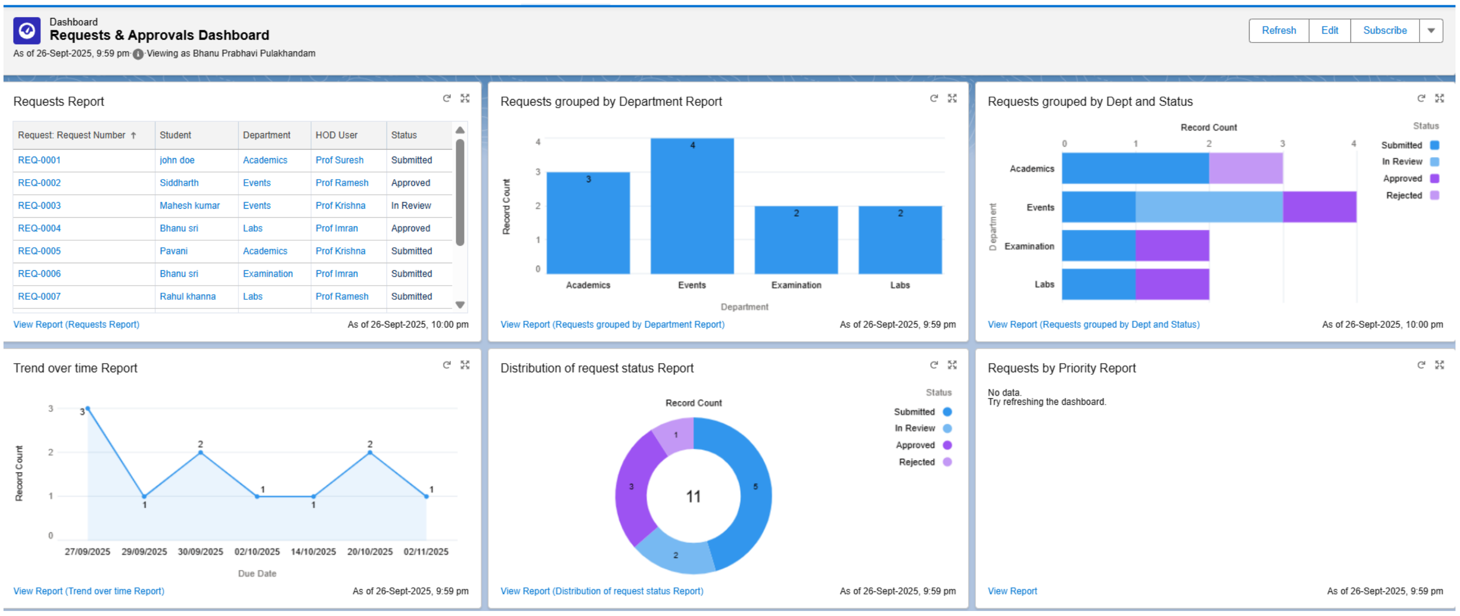
1. **Priority Overview Component**
   - Based on the *Requests by Priority Report*.
   - Displays color-coded metrics for High, Medium, and Low priority requests.
2. **Department Performance Component**
   - Based on *Requests Grouped by Department and Status Report*.
   - Pie charts highlight pending vs approved requests by department.
3. **Trend Analysis Component**

- Based on the *Trend Over Time Report*.
- Line chart showing submission trends for quick identification of peak periods.

4. **Status Distribution Component**
   - Based on the *Distribution of Request Status Report*.
   - Provides an instant snapshot of overall workflow health.

5. **Department Request Volume Table**
   - Based on *Requests Grouped by Department Report*.
   - Gives admins a clear view of workload per department.

6. **Detailed Requests Table**
   - Based on *Requests Report Table*.
   - Allows filtering and sorting for ad-hoc analysis.


Refresh
Edit
Subscribe
Requests grouped by Department Report
Record Count
Department
Academics 3
Events 4
Examination 2
Labs 2
View Report (Requests grouped by Department Report)
As of 26-Sept-2025, 9:59 pm
Requests grouped by Dept and Status
Record Count
Status
Submitted
In Review
Approved
Rejected
Academics
Events
Examination
Labs
View Report (Requests grouped by Dept and Status)
As of 26-Sept-2025, 10:00 pm
Trend over time Report
Record Count
Due Date
27/09/2025
29/09/2025
30/09/2025
02/10/2025
14/10/2025
20/10/2025
02/11/2025
View Report (Trend over time Report)
As of 26-Sept-2025, 9:59 pm
Distribution of request status Report
Record Count
11
View Report (Distribution of request status Report)
As of 26-Sept-2025, 9:59 pm
Requests by Priority Report
View Report
As of 26-Sept-2025, 9:59 pm

Dashboard
Requests & Approvals Dashboard
As of 26-Sept-2025, 9:59 pm · Viewing as Bhanu Prabhavi Pulakhandam

Requests Report

Request: Request Number ↑	Student	Department	HOD User	Status
REQ-0001	john doe	Academics	Prof Suresh	Submitted
REQ-0002	Siddharth	Events	Prof Ramesh	Approved
REQ-0003	Mahesh kumar	Events	Prof Krishna	In Review
REQ-0004	Bhanu sri	Labs	Prof Imran	Approved
REQ-0005	Pavani	Academics	Prof Krishna	Submitted
REQ-0006	Bhanu sri	Examination	Prof Imran	Submitted
REQ-0007	Rahul khanna	Labs	Prof Ramesh	Submitted

View Report (Requests Report) As of 26-Sept-2025, 10:00 pm

No data.
Try refreshing the dashboard.

3. Security Review

Ensuring proper access to reports and dashboards was crucial.

Implemented Measures:

1. **Role-Based Access**
   - Students can view only **reports of their own requests**.
   - Department Heads see **reports and dashboards for their department only**.
   - Admins have full visibility across all departments.

2. **Profile & Permission Sets**
   - Permissions configured to allow access to only relevant reports, dashboards, and underlying objects.
3. **Sharing Rules & OWDs**
   - Organization-Wide Defaults set to private for Request and Student objects.
   - Sharing rules applied to allow department heads visibility into their assigned requests.

Phase 10: Quality Assurance Testing

1. Test Cases

Comprehensive test cases were created to cover every functional aspect of the system.

Key Test Areas:

1. **Request Submission**
   - Students can submit **Leave, Event, Lab, or Project requests**.
   - Verified mandatory fields (Name, Department, Request Type, Notes).
   - Confirmed record creation triggers flows and assigns the request correctly.
2. **Auto-Assignment Flow**
   - Tested Record-Triggered Flow to automatically assign requests to the correct department.
   - Verified correct assignment for all request types.
   - Confirmed handling of bulk request submissions.
3. **Approval and Rejection**
   - Validated department head actions for approving or rejecting requests.
   - Checked status updates and email notifications to students.
4. **Email Notifications**

    - Verified timely delivery of email notifications for approvals, rejections, and escalations.
    - Tested different email templates for correctness.

5. **Escalation Rules**
    - Simulated requests pending >48 hours.
    - Verified that the system automatically escalated the requests and notified administrators.

2. User Acceptance Testing (UAT)

After internal QA, **students and department heads** tested the portal to validate usability and workflow efficiency.

Activities Conducted:

- Students submitted test requests and tracked status updates.
- Department heads approved, rejected, and escalated requests in real-time.
- Users tested navigation, dashboards, and report access.
- Feedback on minor UI improvements, flow adjustments, and notifications was collected.
-

3. Bug Fixes

- Fixed issues in flow auto-assignment.
- Resolved email notification delays.
- Ensured batch Apex jobs run as scheduled.

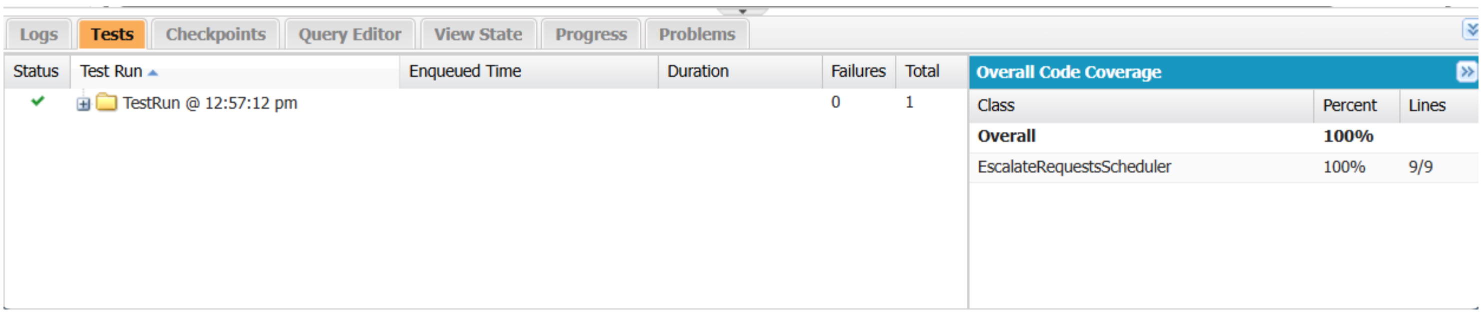
4. Final Verification

A **final verification** was conducted to ensure system readiness:

- **End-to-End Functionality**
    - Student request submission → auto-assignment → approval/rejection → email notification → escalation.
- **Reports & Dashboards**

- Verified dynamic updating of all reports and the consolidated dashboard.
- Ensured metrics reflect real-time data across departments.

- **User Roles & Security**
  - Checked role-based access for students, department heads, and admins.
  - Confirmed sensitive data is visible only to authorized users.



Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 12:57:12 pm			0	1

Overall Code Coverage

Class	Percent	Lines
Overall	**100%**	
EscalateRequestsScheduler	100%	9/9

Executive Summary

Conclusion

The **Student Help & Permission Portal** was designed and implemented on Salesforce to streamline the **request management process between students and departments.** The project successfully advanced through structured phases, including Org Setup, Data Modeling, Process Automation, Apex Programming, User Interface Development, Integration, Data Management, Reporting, and Quality Assurance Testing. Each phase contributed to building a robust solution where students can easily submit and track requests, department heads can efficiently review and approve them, and administrators can monitor institutional performance through real-time dashboards. Automation ensures timely routing and escalation, while custom Lightning components enhance usability. With secure deployment, data integrity, and extensible architecture, the portal transforms the traditional request process into a **centralized, digital, and user-friendly experience**.

Future Scope

The portal has the potential to evolve into a **comprehensive digital campus platform** through:

- **Mobile App Integration** for anytime access.
- **AI-Powered Prioritization** and predictive dashboards.
- **Chatbot & Voice Assistant** support for FAQs and guided submissions.

- **ERP/HRMS Integration** to unify institutional systems.
- **Enhanced Security & Compliance** (MFA, GDPR/FERPA).
- **Cross-Department Expansion** into Finance, Library, and Administration.
- **Gamification Features** to boost engagement and responsiveness.

Recommendations

To ensure sustainability and continuous improvement, the following are recommended:

- **Regular Monitoring & Maintenance** with backups and audits.
- **User Training & Documentation** for students, staff, and admins.
- **Feedback Loop** to gather insights for iterative enhancement.
- **Scalability Planning** to support higher data and user volume.
- **Disaster Recovery Strategy** for business continuity.

- P. Bhanu Prabhavi